GARRLab e primi risultati: automazione e auditing

Luca Vanni
ICRANET
luca.vanni@icranet.org

WORKSHOP GARR 2020

NET MAKERS

Consortium GARR

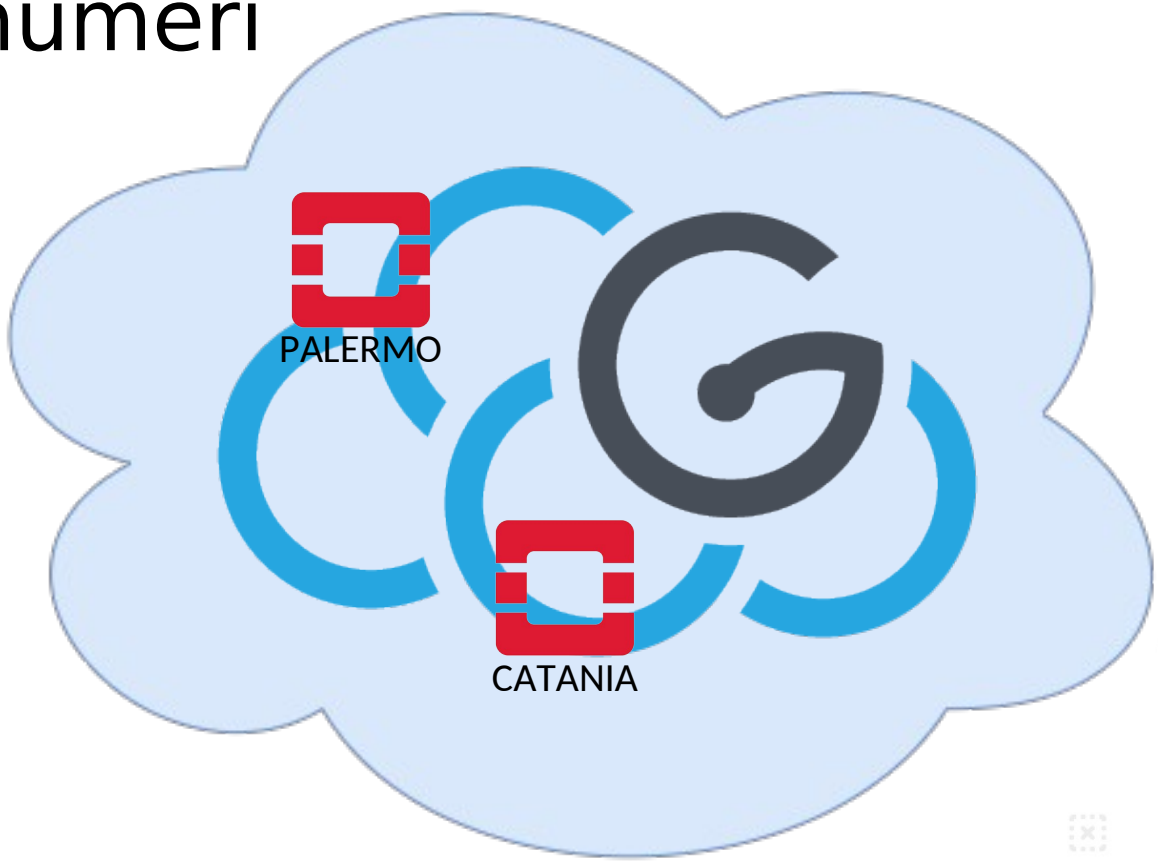
ICRANet

Di cosa parleremo

- Breve overview dell’infrastruttura GarrLab
- Modalità di accesso ai sistemi
  - Uso esclusivo di chiavi (RSA e DSA) per l’autenticazione
  - Gestione della “Privilege escalation”
- Cenni di automazione, deploy utenti e configurazione sudo
- Cenni di auditing, con focus sulla tracciabilità della “Privilege escalation”

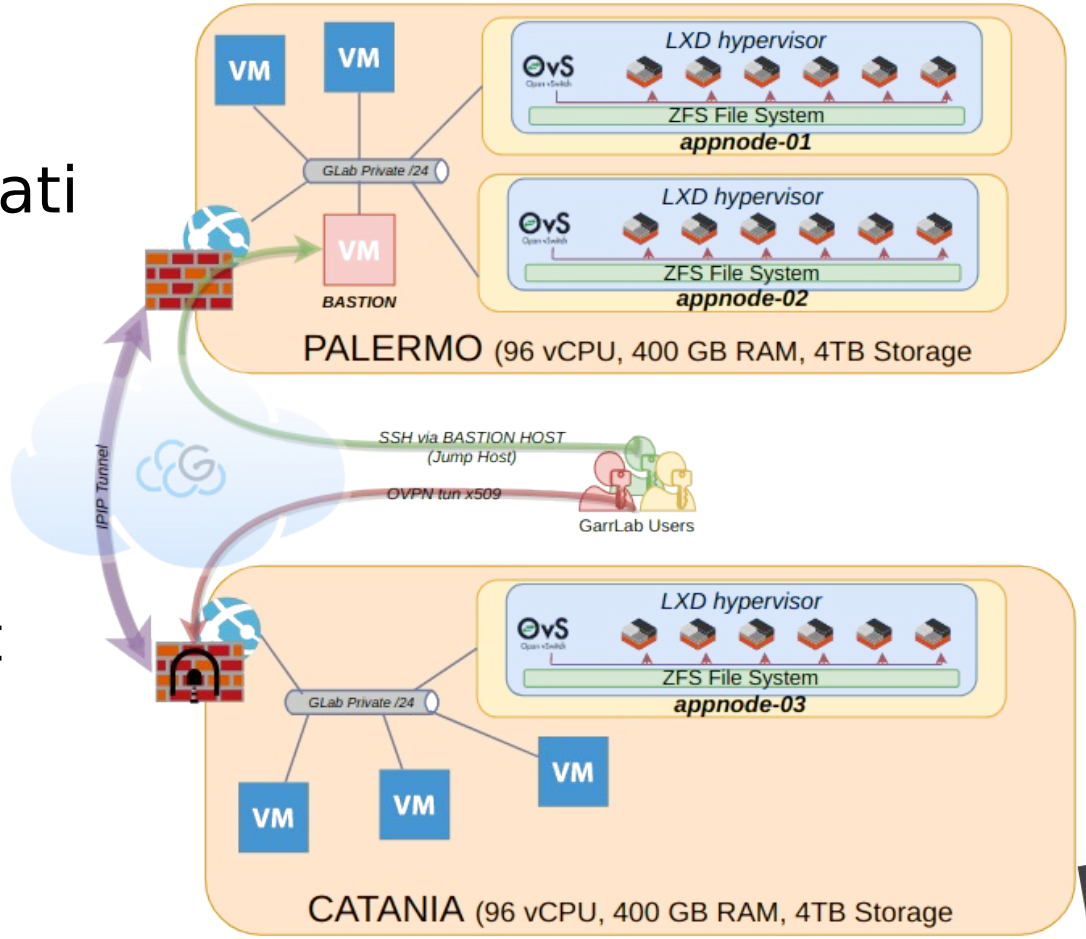
GarrLab: Alcuni numeri

- 2 siti geografici
- 192 Virtual CPU
- 800 GB RAM
- 8 TB Storage



GarrLab: architettura di massima

- 100% servizi erogati via LXC
- Credenziali nominative e password-less
- Full routed subnet (PA – CT)



GarrLab: quali servizi

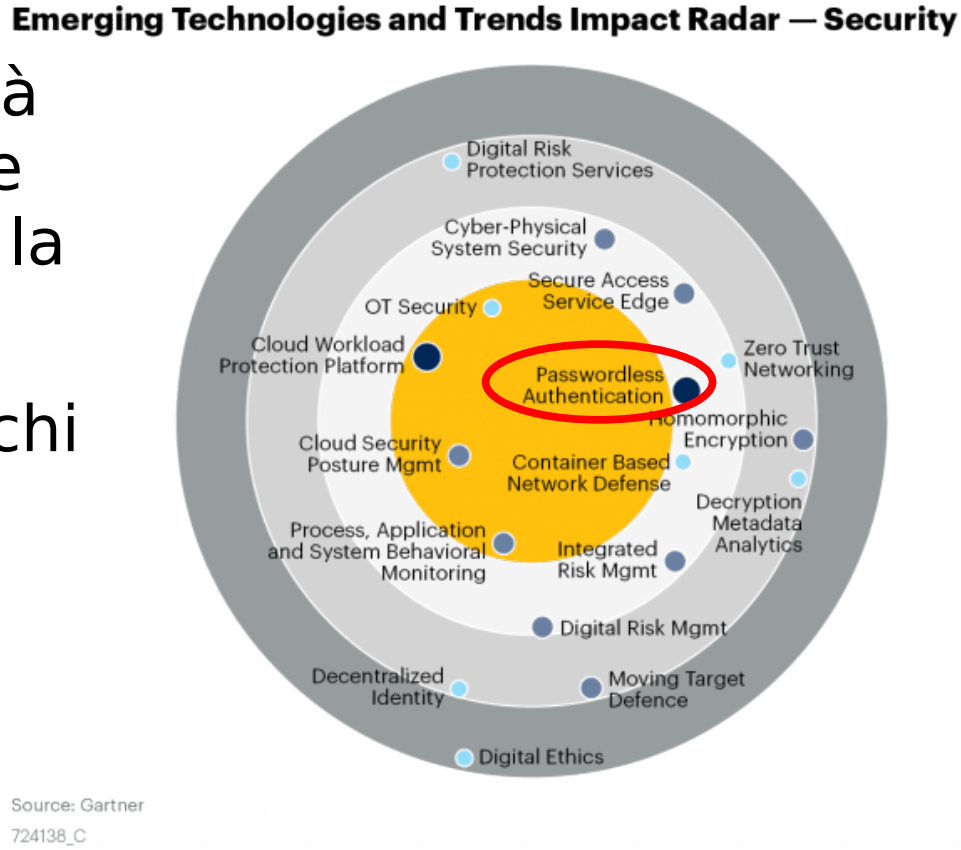
- Strumenti di base (DHCP, DNS, DBMS, etc.)
- GitLab (come strumento di knowledge-sharing)
- Cluster ElasticSearch
- Gateway Telegram
- Progetti GarrLab Up and Running:
  - Wazuh
  - URL Shortener
  - log-biter

Accesso SSH e automazione

- L’accesso ai sistemi è esclusivamente consentito con chiave RSA/DSA
- La chiave pubblica viene univocamente associata alla persona che l’ha fornita
- Il repository delle chiavi è contenuto all’interno dell’host di management dell’infrastruttura (bastion)
- Le chiavi (così come gli utenti) vengono gestite e distribuite via Ansible

SSH & approccio passwordless: Why?

- SSH come modalità di accesso comune ai sistemi su tutta la comunità
- Soggetto ad attacchi brute-force
- Password-policy a volte deboli



https://blogs.gartner.com/swati-rakheja/2020/10/27/announcing-gartners-new-impact-radar-for-security/

Accesso SSH e automazione: Playbook

```yaml
- hosts: ALL-LXC, ALL-GW, ALL-VM-DEVEL, ALL-APPNODE, BASTION-HOSTS
  become: yes
  become_user: root
  become_method: sudo
  handlers:
    [...]
  tasks:
    [...]
    - name: Create group 'lvanni'
      group:
        name: lvanni
        gid: 2001
    - name: Add/Set user 'lvanni' with a specific uid and a primary group of 'lvanni'
      user:
        name: lvanni
        comment: Luca Vanni
        uid: 2001
        group: lvanni
        groups: "lvanni, garrdevs"
        home: "/home/lvanni"
        shell: "/bin/bash"
        append: yes
    - name: Deploy SSH Key for user 'lvanni'
      authorized_key:
        user: lvanni
        state: present
        key: "{{ lookup('file', '/opt/devops_repo/ansible/assets/ssh-keys/lvanni_ssh_key.pub') }}"
    [...]
```

- 100% infrastruttura GarrLab coperta (~ 35 host)
- Playbook “idempotente”

Privilege escalation con sudo

- `sudo` passwordless riutilizzando ESCLUSIVAMENTE la chiave pubblica dell'utente con la quale ha eseguito l'accesso al sistema *(integrando la libreria pam_ssh_agent_auth)*
- Controllo delle autorizzazioni ad ogni comando eseguito con privilegi elevati (no `sudo` timeout)
- Necessaria riconfigurazione del client SSH per abilitare l'agent forwarding

Privilege escalation con sudo: Playbook

```yaml
- hosts: ALL-LXC, ALL-GW, ALL-VM-DEVEL, ALL-APPNODE, BASTION-HOSTS
  become: yes
  become_user: root
  become_method: sudo
  handlers:
    [...]
  tasks:
    [...]
    - name: installing libpam-ssh-agent-auth
      apt: name=libpam-ssh-agent-auth state=latest
    - name: Updating 'sudoers' config to preserve the environment variable SSH_AUTH_SOCK
      lineinfile:
        path: /etc/sudoers
        line: 'Defaults env_keep += "SSH_AUTH_SOCK"'
        insertbefore: Defaults.*
    - name: Configure 'sudo' to try using public keys
      blockinfile:
        path: /etc/pam.d/sudo
        block: |
                auth sufficient pam_ssh_agent_auth.so file=~/.ssh/authorized_keys
        insertafter: PAM.*
    [...]
```

```yaml
    [...]
    - name: Remove 'sudo' timeout
      replace:
        path: /etc/sudoers
        regexp: '(.*env_reset.*)'
        replace: 'Defaults env_reset, timestamp_timeout=00'
    - name:  Delete template sudoers file
      file:
        path: /etc/sudoers.d/90-cloud-init-users
        state: absent
    - name: Updating 'sudoers' config for all GLOBAL sysadmin user
      blockinfile:
        path: /etc/sudoers.d/50-garrcloud-GLOBAL-ADMIN
        create: yes
        force: yes
        block: |
                lvanni ALL=(ALL) ALL
                dverzulli ALL=(ALL) ALL
                eardizzoni ALL=(ALL) ALL
    - name: Disable sudo password auth
      replace:
        path: /etc/pam.d/sudo
        regexp: '(.*@include common-auth.*)'
        replace: '# @include common-auth'

  [...]
```

Privilege escalation con sudo: risultato

```
#%PAM-1.0
# BEGIN ANSIBLE MANAGED BLOCK
auth sufficient pam_ssh_agent_auth.so file=~/.ssh/authorized_keys
# END ANSIBLE MANAGED BLOCK

session    required   pam_env.so readenv=1
session    required   pam_env.so readenv=1
# @include common-auth
@include common-account
@include common-session-noninteractive
~
~
"/etc/pam.d/sudo" [readonly] 10L, 365C
```

```
# BEGIN ANSIBLE MANAGED BLOCK
lvanni ALL=(ALL) ALL
dverzulli ALL=(ALL) ALL
eardizzoni ALL=(ALL) ALL
# END ANSIBLE MANAGED BLOCK
~
~
"/etc/sudoers.d/50-garrcloud-GLOBAL-ADMIN" 5L, 128C
```

- 100% Ansible managed

Auditing: accessi SSH e Privilege escalation

```
2020-10-29T21:15:46.380570+01:00 munin sshd[6048]: Accepted publickey for lvanni from 172.16.16.254 port 50436 ssh2: RSA SHA256:Ii74ORZDxd5bY8iZc0QijBTZ4JZjZaL9aJd4eOaduE
2020-10-29T21:15:46.382842+01:00 munin sshd[6048]: pam_unix(sshd:session): session opened for user lvanni by (uid=0)
2020-10-29T21:15:46.405941+01:00 munin systemd-logind[213]: New session 309239 of user lvanni.
2020-10-29T21:15:46.407414+01:00 munin systemd: pam_unix(systemd-user:session): session opened for user lvanni by (uid=0)
2020-10-29T21:15:51.517997+01:00 munin sudo[6174]: pam_ssh_agent_auth: matching key found: file/command /home/lvanni/.ssh/authorized_keys, line 1
2020-10-29T21:15:51.518520+01:00 munin sudo[6174]: pam_ssh_agent_auth: Found matching RSA key: 2b:cd:04:4c:3d:76:46:be:09:07:da:90:7d:1e:9e:20
2020-10-29T21:15:51.729990+01:00 munin sudo[6174]: pam_ssh_agent_auth: Authenticated: `lvanni' as `lvanni' using /home/lvanni/.ssh/authorized_keys
2020-10-29T21:15:51.757917+01:00 munin sudo:   lvanni : TTY=pts/0 ; PWD=/home/lvanni ; USER=root ; COMMAND=/bin/bash
2020-10-29T21:15:51.759086+01:00 munin sudo: pam_unix(sudo:session): session opened for user root by lvanni(uid=0)
```

Auditing: accessi SSH e Privilege escalation

```
2020-10-29T21:15:46.380570+01:00 munin sshd[6048]: Accepted publickey for lvanni from 172.16.16.254 port 50436 ssh2: RSA SHA256:/Ii74ORZDxd5bY8iZc0QijBTZ4JZjZaL9aJd4eOaduE
2020-10-29T21:15:46.382842+01:00 munin sshd[6048]: pam_unix(sshd:session): session opened for user lvanni by (uid=0)
2020-10-29T21:15:46.405941+01:00 munin systemd-logind[213]: New session 309239 of user lvanni.
2020-10-29T21:15:46.407414+01:00 munin systemd: pam_unix(systemd-user:session): session opened for user lvanni by (uid=0)
2020-10-29T21:15:51.517997+01:00 munin sudo[6174]: pam_ssh_agent_auth: matching key found: file/command /home/lvanni/.ssh/authorized_keys, line 1
2020-10-29T21:15:51.518520+01:00 munin sudo[6174]: pam_ssh_agent_auth: Found matching RSA key 2b:cd:04:4c:3d:76:46:be:09:07:da:90:7d:1e:9e:20
2020-10-29T21:15:51.729990+01:00 munin sudo[6174]: pam_ssh_agent_auth: Authenticated: `lvanni' as `lvanni' using /home/lvanni/.ssh/authorized_keys
2020-10-29T21:15:51.757917+01:00 munin sudo:   lvanni : TTY=pts/0 ; PWD=/home/lvanni ; USER=root ; COMMAND=/bin/bash
2020-10-29T21:15:51.759086+01:00 munin sudo: pam_unix(sudo:session): session opened for user root by lvanni(uid=0)
```

```
luca@XPS-LV:~$ ssh-keygen -lf public_key_LV -E sha256
8096 SHA256:/Ii74ORZDxd5bY8iZc0QijBTZ4JZjZaL9aJd4eOaduE rsa-key-LV (RSA)
luca@XPS-LV:~$ ssh-keygen -lf public_key_LV -E md5
8096 MD5:2b:cd:04:4c:3d:76:46:be:09:07:da:90:7d:1e:9e:20 rsa-key-LV (RSA)
luca@XPS-LV:~$
```

- Siamo “sicuri” che l’utente che ha effettuato l’accesso è lo stesso che ha richiesto l’esecuzione di un comando con privilegi elevati

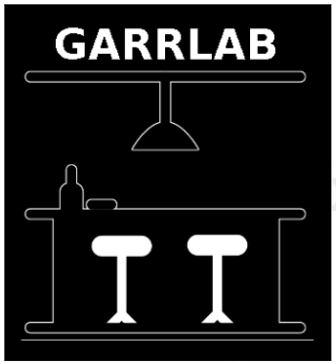
Conclusioni

- L’approccio mostrato rappresenta soltanto “un passo” nel lungo cammino verso una piattaforma di auditing di livello enterprise (auditd, selinux, etc.);
- È comunque fortemente consigliato allestire preliminarmente una infrastruttura di raccolta ed archiviazione dei LOG (su host off-site e parzialmente inaccesibili)
- Se possibile, è opportuno affiancare una piattaforma di analisi dei propri log con l’impiego di sistemi “intelligenti”

JOIN US!

https://gitlab.garrlab.it/ansible

https://www.garrlab.it



Grazie per l’attenzione