VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINIŲ SISTEMŲ KATEDRA

DUOMENŲ ATSARGINĖS KOPIJOS/SIEM

Namų darbas nr. 4

Atliko: ITSfm-22 grupės studentas Aurimas Šakalys
Tikrino: lektorius Vitalijus Gurčinas

Vilnius, 2022

Turinys

1 Duomenų atsraginės kopijos **3**
1.1 Programinės įrangos pasirinkimas 3
1.2 Programinės įrangos diegimas ir konfiguravimas 3
1.3 Užduoties reikalavimų įgyvendinimas 5

2 SIEM **5**
2.1 Programinės įrangos pasirinkimas 5
2.2 Programinės įrangos diegimas ir konfiguravimas 6
2.3 Užduoties reikalavimų įgyvendinimas 7

Iliustracijų sąrašas

1 *borg* repozitorijose saugomi archyvai 5
2 *Wazuh* įrankio komponentinė diagrama 6
3 Žurnalinių įrašų agregacijos konfiguracija 7
4 Saugos įvykių nurodytose direktorijose agregacijos konfiguracija 7
5 *CIS* patikros rezultatai *wazuhAgent1* agentui 7

Ištraukų sąrašas

1 SSH rakto apribojimas vienai komandai 3
2 *Borgmatic konfiguracija* 3
3 *cron* įrašas, kas minutę kviečiantis *Borgmatic* 4

1. Duomenų atsraginės kopijos

1.1. Programinės įrangos pasirinkimas

Atsarginėms duomenų kopijoms daryti pasirinkau du įrankius - *borg* ir *Borgmatic*.

borg įrankis skirtas daryti atsargines duomenų kopijas. Įrankis veikia *serverio-kliento* principu, kur serveris atlieka duomenų saugojimo užduočių vykdymą, o klientas - deduplikaciją, šifravimą, suspaudimą ir pan.

Borgmatic įrankis skirtas gerokai palengvinti darbą su *borg* atsarginių kopijų darymo įrankiu. Šis įrankis suteikia vartotojui paprastą vartotojo sąsają ir konfiguraciją. Įrankį galima sukonfiguruoti įvariapusiškai, įjungiant ar išjungiant funkcionalumą, kurį suteikia *borg* įrankis.

1.2. Programinės įrangos diegimas ir konfiguravimas

Virtualios mašinos kuriamos naudojant *Vagrant* įrankį, o pačios mašinos yra automatiškai sukonfiguruojamos naudojant *Ansible* pjesėmis.

Norint įgyvendinti pirmąjį namų darbo reikalavimą, nusprendžiau atlikti kopijų darymą naudojant **3-1-2** strategiją. Ši strategija įgyvendinta lokaliai sukuriant dvi virtualias mašinas, į kurias bus talpinamos atsarginės kopijos. Nors praktine prasme abi *borg* serverio virtualios mašinos egzistuoja toje pačioje vietoje, šias virtualias mašinas galima būtų paleisti ir sukonfiguruoti debesų kompiuterijos infrastruktūroje, taip abu atsarginių kopijų serverius patalpinant skirtinguose vietose. Taip **3-1-2** strategija būtų įgyvendinama praktiškai.

Prieš pradedant atsarginių kopijų sistemos diegimą, klientinėje mašinoje privaloma sugeneruoti SSL raktų porą, kuri bus naudojama autentifikuojantis siunčiant duomenis į atsarginių kopijų serverį. Kiekvieno kliento raktas turi būti įdiegtas į atsarginių kopijų darymo serverius. Diegiant viešąjį kliento raktą, šis raktas yra apribojamas *tik* atsarginių kopijų darymo tikslams pridedant šį apribojimą matomą Ištrauka 1. Tokiu būdu užtikrinama, kad SSL raktų pora negalės būti panaudota kitiems tikslams serveryje. Tolimesni žingsniai apibūdinami tariant, jog tarp serverio ir kliento galima užmegzti saugų SSH ryšį.

Ištrauka 1: SSH rakto apribojimas vienai komandai

```
1 command="borg serve --restrict-to-path /var/local/backups/backup.
    borg",restrict MIIDDjCCAfY...
```

Serverinėse mašinose paprasčiausiai yra įdiegiama *borg* programinė įranga, kuri leis atlikti kopijų darymą klientinei mašinai.

Klientinėje mašinoje, įdiegiami *borg* ir *Borgmatic* įrankiai. *Borgmatic* įrankis yra sukonfiguruojamas naudojant konfiguracinį failą matomą Ištrauka 2.

Ištrauka 2: *Borgmatic konfiguracija*

```
1 location:
2     source_directories:
```

```
        - /home
        - /etc
        - /var/log/syslog*
    repositories:
        - ssh://vagrant@192.168.0.202/var/local/backups/backup.
          borg
        - ssh://vagrant@192.168.0.203/var/local/backups/backup.
          borg
        - /var/local/backups/backup.borg

storage:
    encryption_passphrase: "encryption_password"
    ssh_command: ssh -i /home/vagrant/.ssh/id_rsa -o
      StrictHostKeyChecking=accept-new

retention:
    keep_minutely: 60
    keep_daily: 7
    keep_monthly: 6
    keep_yearly: 1
```

Konfiguracija nurodo:

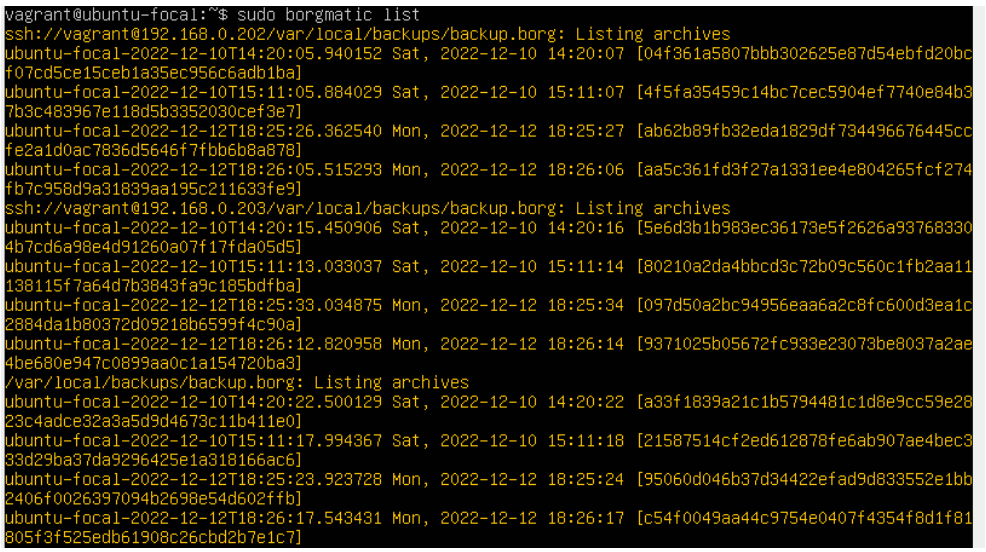
- **source_directories** - kurioms direktorijoms/failams norime sudaryti atsarginių duomenų kopijas;
  - Šiuo atveju sudarysime direktorijų/failų */home*, */etc* ir */var/log/syslog** atsargines kopijas.
- **repositories** - į kurias *borg* repozitorijas norėtume talpinti atsargines kopijas;
  - Šiuo atveju, dvi atsarginės kopijos talpinamos į du atsarginių kopijų serverius, viena talpinama lokaliai.
- **storage** - nurodo šifravimo slaptažodį ir komandą, naudojamą prisijungti prie atsraginių kopijų serverių;
- **retention** - nurodo kiek, kokio tipo atsarginių kopijų archyvų paliksime. [1]

Atlikus įrankio konfiguraciją telieka įdiegti *cron* įrašą (Ištrauka 3), kuris kas tam tikrą laiką pakviestų *Borgmatic* įrankį, taip atliekant duomenų kopijų sudarymą ir išsaugojimą.

Ištrauka 3: *cron* įrašas, kas minutę kviečiantis *Borgmatic*

```
    */1 * * * * root PATH=$PATH:/usr/bin:/usr/local/bin /usr/
      local/bin/borgmatic --verbosity -1 --syslog-verbosity 1
```

[1] Norėčiau atkreipti dėmesį, jog visu 100% nesuprantu, kokia metodologija yra išlaikomi deduplikuoti duomenys po senų archyvų pašalinimo. *borg* įrankis savo dokumentacijoje šio mechanizmo neaprašo.



1 pav. *borg* repozitorijose saugomi archyvai

1.3. Užduoties reikalavimų įgyvendinimas

Atsarginės kopijos atliekamos daromos naudojant **3-1-2** strategiją, taip įgyvendinant pirmąjį reikalavimą. Kadangi *borg* įrankis yra deduplikuojantis pagal nutylėjimą ir yra padaromos tik pakeistų ar naujų failų atsarginės kopijos, o duomenys yra šifruojami, yra įgyvendinami antras ir penktasis reikalavimai. *borg* įrankis sudaro šifruotas ir inkrementalias atsargines kopijas, taip apsisaugant nuo išpirkos atakų ir įgyvendinant ketvirtą reikalavimą.

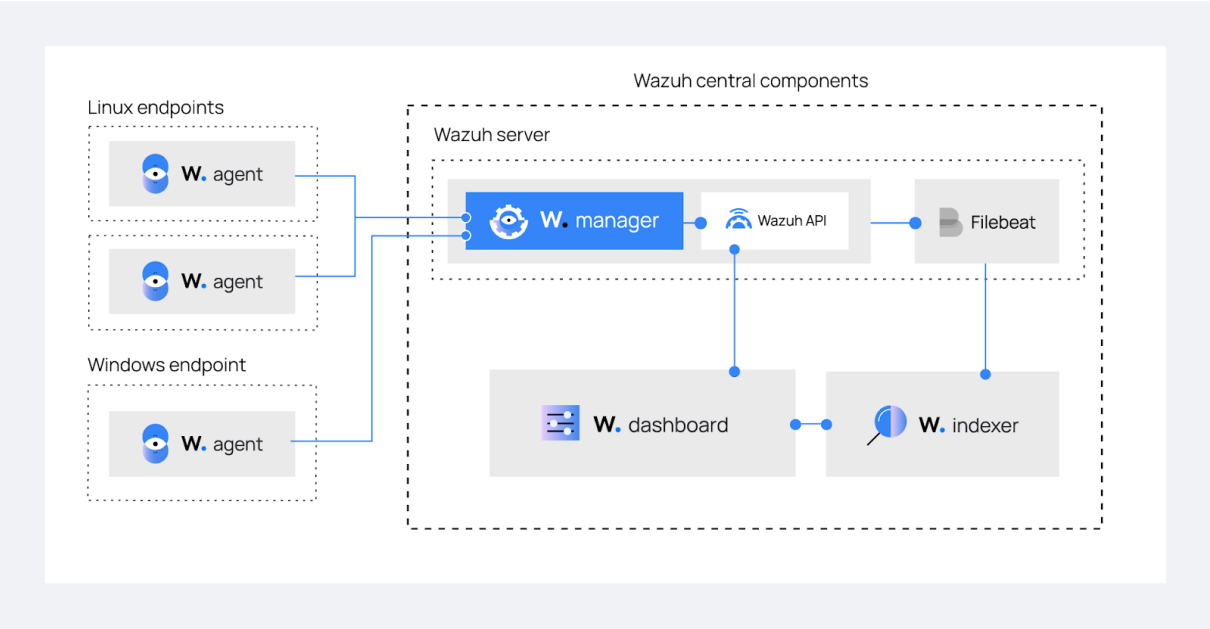
Trečiasis reikalavimas įgyvendinamas kiek sudetingiau. Kadangi yra paliekami dienos (septynios), savaitės (keturios) ir mėnesio (šešios) atsarginės kopijos, duomenis galima atkurti naudojant juos, tačiau šie nebūtinai gali vadintis pilnomis atsarginių duomenų kopijomis. Norint pilnai atitikti reikalavimą, kiekvienos savaitės pilnam archyvui tektų sukurti po *borg* repozitoriją. Tai nėra patogu, tačiau tai užtikrintų šimtąprocentinį reikalavimo įgyvendinimą.

Atsarginės duomenų kopijos yra daromos automatiškai, todėl įgyvendinamas ir papildomas reikalavimas.

2. SIEM

2.1. Programinės įrangos pasirinkimas

SIEM sistemos įgyvendinimui pasirinkau *Wazuh* įrankį dėl paprasto diegimo ir naudojimo.



2 pav. *Wazuh* įrankio komponentinė diagrama

Wazuh įrankis susideda iš kelių komponentų. *Wazuh agent* yra komponentas, atsakingas už saugos įvykių registravimą ir žurnalinių įrašų stebėjimą klientinėje mašinoje. Šis komunikuoja su *Wazuh manager* komponentu, kuris perduoda gautus duomenis iš klientinių mašinų indeksatoriui. Besinaudojant indeksatoriaus duomenimis *Wazuh dashboard* vartotojo sąsajos komponentas atvaizduoja šiuos duomenis įvarus tipo diagramose ir pan.

2.2. Programinės įrangos diegimas ir konfiguravimas

Virtualios mašinos kuriamos naudojant *Vagrant* įrankį, o pačios mašinos yra automatiškai sukonfiguruojamos naudojant *Ansible* pjesėmis.

Naudojamos dvi virtualios mašinos, viena - *Wazuh* serveris, kita - *Wazuh Ubuntu Linux* klientas.

Diegimas vyksta keliais etapais. Pirmojo etapo metu yra sugeneruojami *Wazuh* serverio sertifikatai, šie eksportuojami panaudojimui klientinėse mašinose. Šiuos sugeneravus, įdeigiami serverio komponentai, indeksatorius ir *Wazuh dashboard*, kartu ir *Wazuh manager* su *Filebeat*.

Sudiegus serverį, galima pradėti konfiguruoti klientines mašinas, į jas diegiant *Wazuh agent*. Visi veiksmai atliekami naudojantis *Wazuh* suteikiamomis *Ansible* pjesėmis ir rolėmis.

Konfiguracija kuri sudiegiama naudojantis pateikiamomis pjesėmis yra tvarkinga ir atitinkati didžiąją dalį užduoties reikalavimų. Figure 3 galime matyti jog sukonfiguruojami *syslog* tipo žurnalinių įrašų nuskaitymas iš nurodytų failų. Šie įvykiai yra išsaugomi *Wazuh* serveryje ir juos galima pamatyti *Wazuh dashboard*. Figure 4 galime matyti saugos įvykių generavimo konfiguraciją, šiose direktorijose atlikti veikmai sugeneruotu įvykius, kurie būtų patalpinti serveryje.

Wazuh nesuteikia funkcionalumo automatiškai ištrinti žuranlinius įrašus ar įvykius, tačiau tai galima atlikti naudojant *cronjob* funkcionalumą.

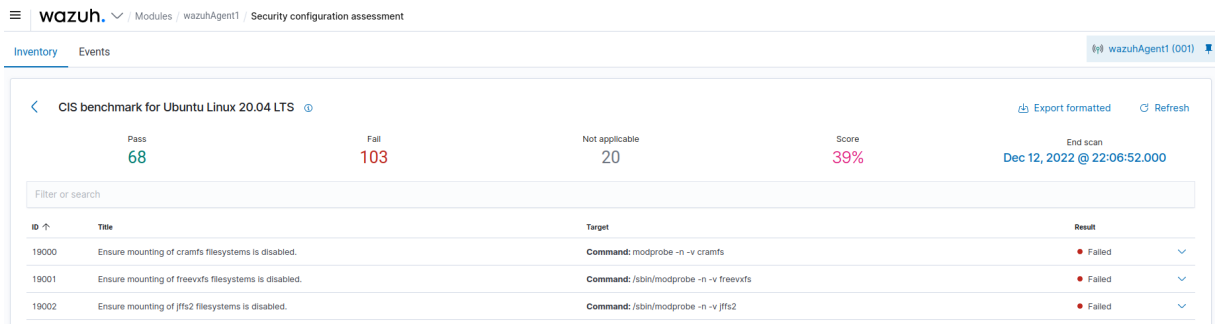
```
<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/auth.log</location>
</localfile>

<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/syslog</location>
</localfile>

<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
</localfile>

<localfile>
    <log_format>syslog</log_format>
    <location>/var/log/kern.log</location>
</localfile>
```

3 pav. Žurnalinių įrašų agregacijos konfiguracija

```
<syscheck>
  <disabled>no</disabled>
  <frequency>43200</frequency>
  <scan_on_start>yes</scan_on_start>
  <!-- Directories to check  (perform all possible verifications) -->
  <directories >/etc,/usr/bin,/usr/sbin</directories>
  <directories >/bin,/sbin,/boot</directories>
  <directories check_all="yes" report_changes="yes" realtime="yes">/home/vagrant</directories>
```

4 pav. Saugos įvykių nurodytose direktorijose agregacijos konfiguracija

Sistema taip pat atlieka didelį kiekį įvairaus tipo patikrų, kaip *CIS* ar pažeidžiamumų patikros.



5 pav. *CIS* patikros rezultatai *wazuhAgent1* agentui

2.3. Užduoties reikalavimų įgyvendinimas

Įdiegta *Wazuh* SIEM sistema nuskaito žurnalinius įrašus, generuoja saugumo įvykius ir pranešimus. Taip įgyvendinami pirmi trys užduoties reikalavimai. Penktasis reikalavimas gali būti įgyvendinamas naudojant *crontab* funkcionalumą serveryje.