Veilige backup

binnen een ‘onveilige’ omgeving

Voorstel backup (zaken) inhoud

- Huidige situatie
  - Opzet
  - Nadelen
  - Voordelen
- Ideaal? plaatje
- Werkbaar? plaatje
  - Voorstel asynchrone opslag
    - Eisen / concept opzet
    - Software
    - Nadelen
    - Voordelen
- Demo POC

huidige situatie opzet

- ‘Werk’ opslag
  - Acquisitie bestemming
  - Onderzoek aan images (IEF / Encase / mmls)
  - R/W voor iedereen

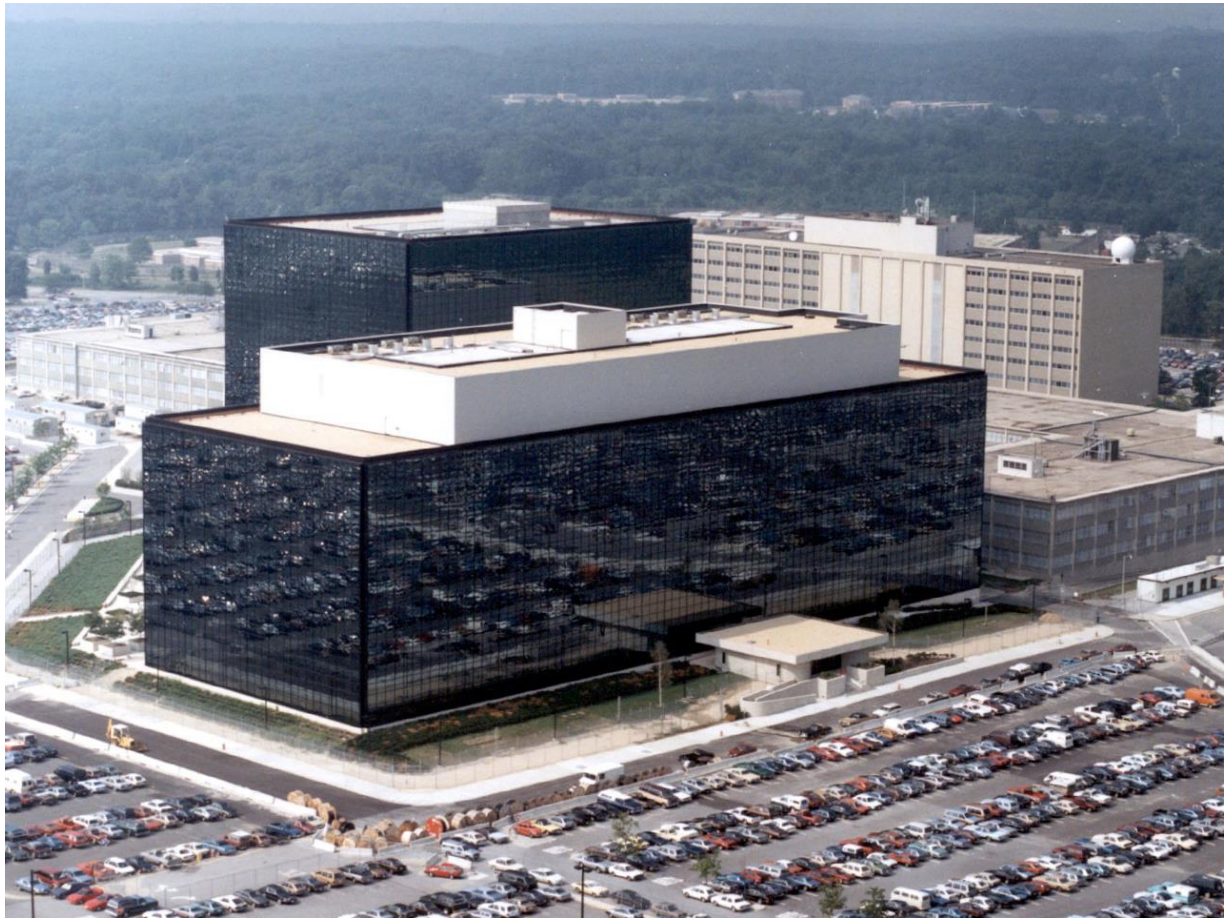
- ‘Backup’ opslag
  - Tweede kopie voor acquisitie
  - Backup in geval dat bij onderzoek een image beschadigd raakt
  - R/W voor iedereen
- Initiatief vanuit gebruiker (/ tool)

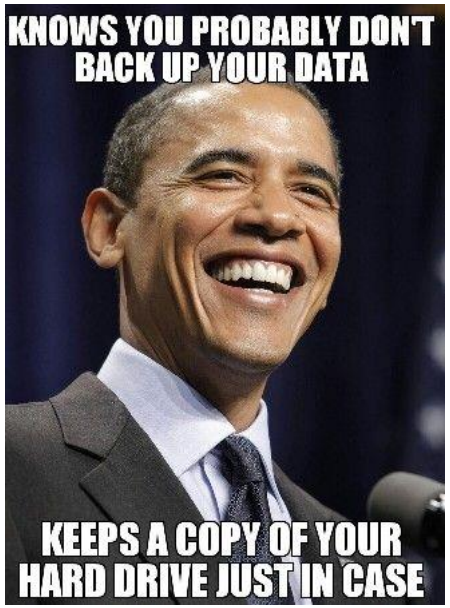
Huidige situatie voor-/nadelen

- KISS
- Eenvoudig actief ←→ backup vergelijken
- Eenvoudig scheve situatie herstellen

- Backup
  - Is online
    - Kwetsbaar voor malware / malfunction / malaction
  - Is benaderbaar en schrijfbaar voor iedereen
  - Heeft geen zeggenschap over integriteit data
    - Heeft het maar te slikken
  - ‘Push’ mechanisme

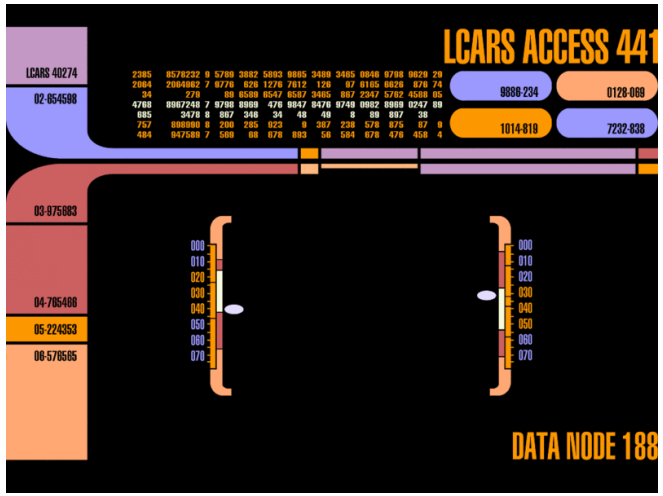
Huidige situatie 3e lijns support





Ideaal plaatje

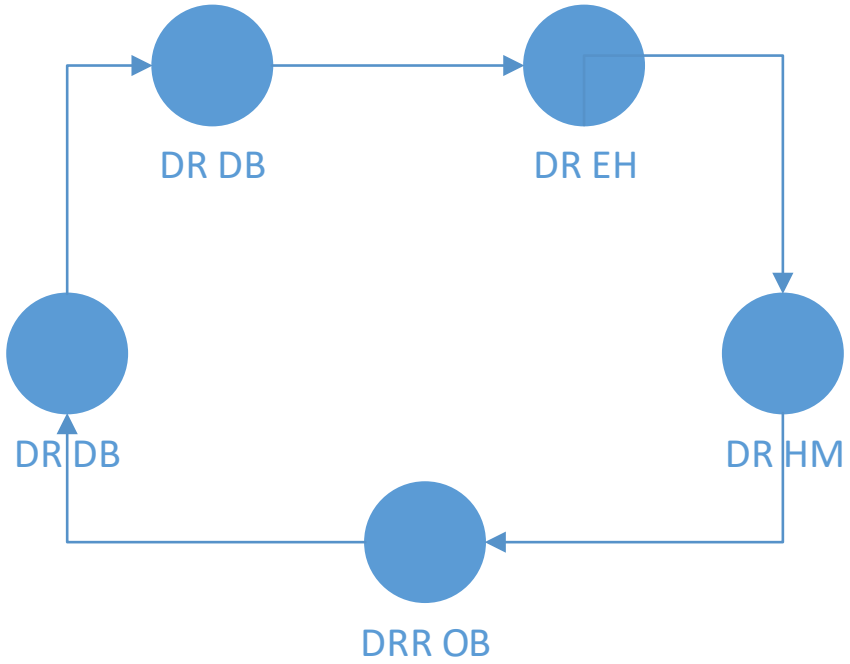
…want Murphy



Ideaal plaatje

- Offsite, want:
  - Groot incident → alles op locatie is verloren
    - b.v. EMP, maar meer waarschijnlijk ☺ brand / instorting / …
- (semi) ‘Offline’ want:
  - Actieve dreiging zal op zoek gaan naar doelen.
  - Denk ransomware
    - (b.t.w. is allang NIET meer alleen SMB shares.
    - (meest recente voorbeeld zijn Mongo DB servers)
- Hoge throughput want:
  - Hoeveelheid data zal waarschijnlijk niet afnemen

Ideaal plaatje



Ideaal plaatje ooit... maar nog niet

- Locaties moeten gekoppeld zijn
- Beveiliging / confidentialiteit data per locatie
- Gecombineerd met oplossing hierna

Voorstel korte termijn eisen

- ‘Onzichtbaar’ op netwerk
- ‘Pull’
  - Interval configureerbaar
- Integriteits controle
- Geen neven activiteiten

Voorstel – opzet conceptueel

- Gebruiker
  - kopieert data naar ‘werk’ opslag
- Werk-opslag
  - (optie: berekent hashes en slaat deze op)
- ...
- Backup-opslag
  - Polling - ziet nieuwe data
  - *Verifieert integriteit werk-opslag adhv controle bestanden
  - *Haalt nieuwe data op
  - Berekent hashes en slaat deze op
  - (haalt hashes nieuwe data van werk-opslag op)
  - (vergelijkt berekende hashes met hashes van werk-opslag)

*‘Verifieert integriteit’

- Blind kopieëren verschuift probleem.
  - Malware corrumpteerd werk-opslag
  - Backup-opslag synct → backup-opslag ook corrupt
- Optie - versionering in bestandssysteem
  - b.v. ZSF
- Optie - integriteits controle werk-opslag
  - Uitgevoerd door backup-opslag
  - Referentiebestanden op werk-opslag, hashes op backup-opslag
    - Verschillende type, ‘echte’, bestanden (.docx, xml, html, mp4, e01, ...)
  - Backup-opslag haalt validatie bestanden op en verifieert hash
  - Bij afwijking → Alarm

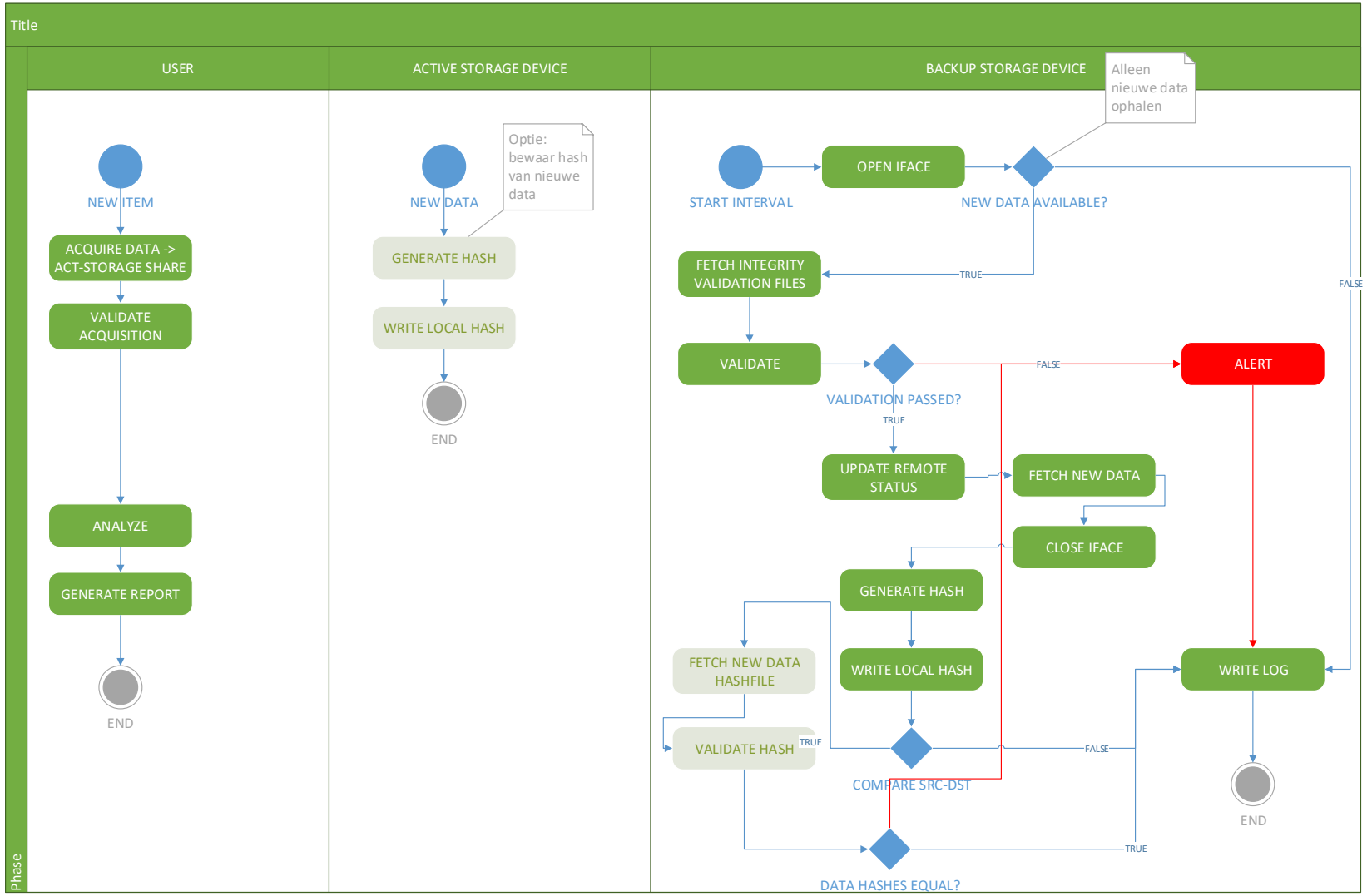
*‘Haalt nieuwe data op’ # 1

- Wat is ‘nieuw’?
- Tijdelijke bestanden / gegenereerde bestanden / ...
  - Hoeven niet per-sé mee
- Malversief gemuteerde bestaande bestanden
  - MOGEN NIET MEE

*‘Haalt nieuwe data op‘ # 2

- Optie 1 – detecteer en alarmeer
  - Data al aanwezig → Infor-/Alarmeer en sla over
  - ‘forceer’ signaalbestand om bewust bestaande data bij te werken
  - Optie: combineer met centraal / lokaal ‘negeer’ bestand
    - Overslaan viewers / tijdelijke bestanden / known-gegenereerde databestanden
- Optie 2 – verifieer data in tijdelijke plek
  - Is een document leesbaar, een E01 geldig, etc...
    - B.v. libewf / ...
  - Blijven gaten
  - Kan gecombineerd worden met optie 1

Voorstel Opzet conceptueel



Voorstel Implementatie

- Omgeving
  - Multiplatform
  - POC gebaseerd op Linux
- Utilities
  - ssh/scp
  - Rsync (over ssh, mogelijk niet nodig ←→ performance)
  - CRON (Systemd-timer)
  - Iptables / ifconfig (/ netsh)
- Script
  - Python

Voorstel Implementatie (code)

- subprocess.run([‘rm’, ‘rf’, ‘/data’], stdout = subprocess.PIPE)
- subprocess.run([‘cp’, ‘fr’, ‘backup@frivo:/’, ‘/data’])

- Simpel / leesbaar / fool-proof

- OFTEWEL: <walkthrough CODE runner_backup.py>

Voorstel implementatie - TODO

- https://github.com/mgdegroot/bastard
  - (Backup Acquire and Signing Tool for Assorted Repositories of Data)
- Unit tests / systeemtest / SAT
- Documentatie
- Configuratie
  - Gewenst standaard gedrag
  - Config file ipv arguments
- Fout afhandeling
  - robuustheid

Voorstel Demo

- nas1 – werk opslag
  - IP 192.168.42.10
  - Gebruiker ‘backup’ met lees / schrijfrechten op share
  - Samba smbd / nmbd share
    - ‘Onderzoeken’ → /media/storage/Onderzoeken
- nas2 – backup opslag
  - IP 192.168.42.11
  - Opslag locatie /media/storage
    - ./backup → data
    - ./meta → log
    - ./validation → test bestanden

Voorstel Demo

Schietkwartier

a.k.a.: vragen?