Automatizacija pripreme kiberbetičkih poligona

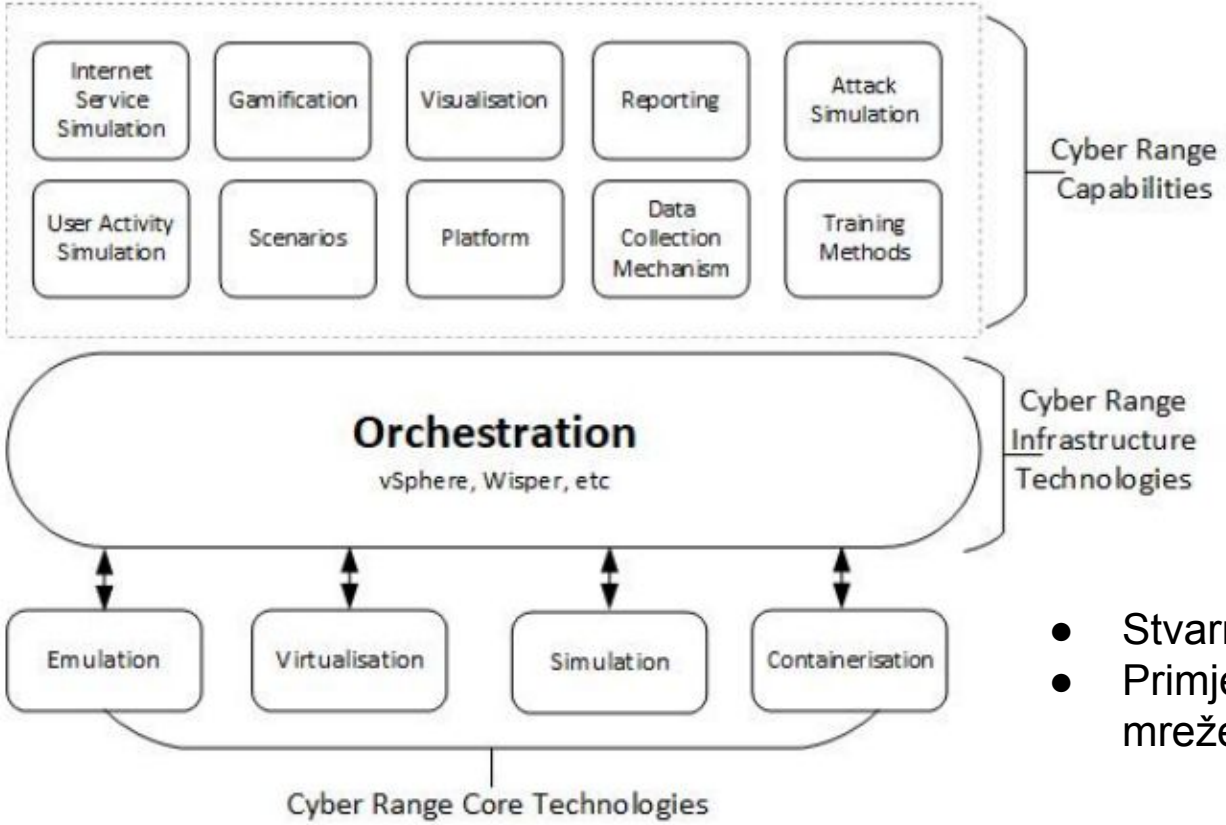
Ivan Kovačević

Sadržaj

Kibernetički poligoni

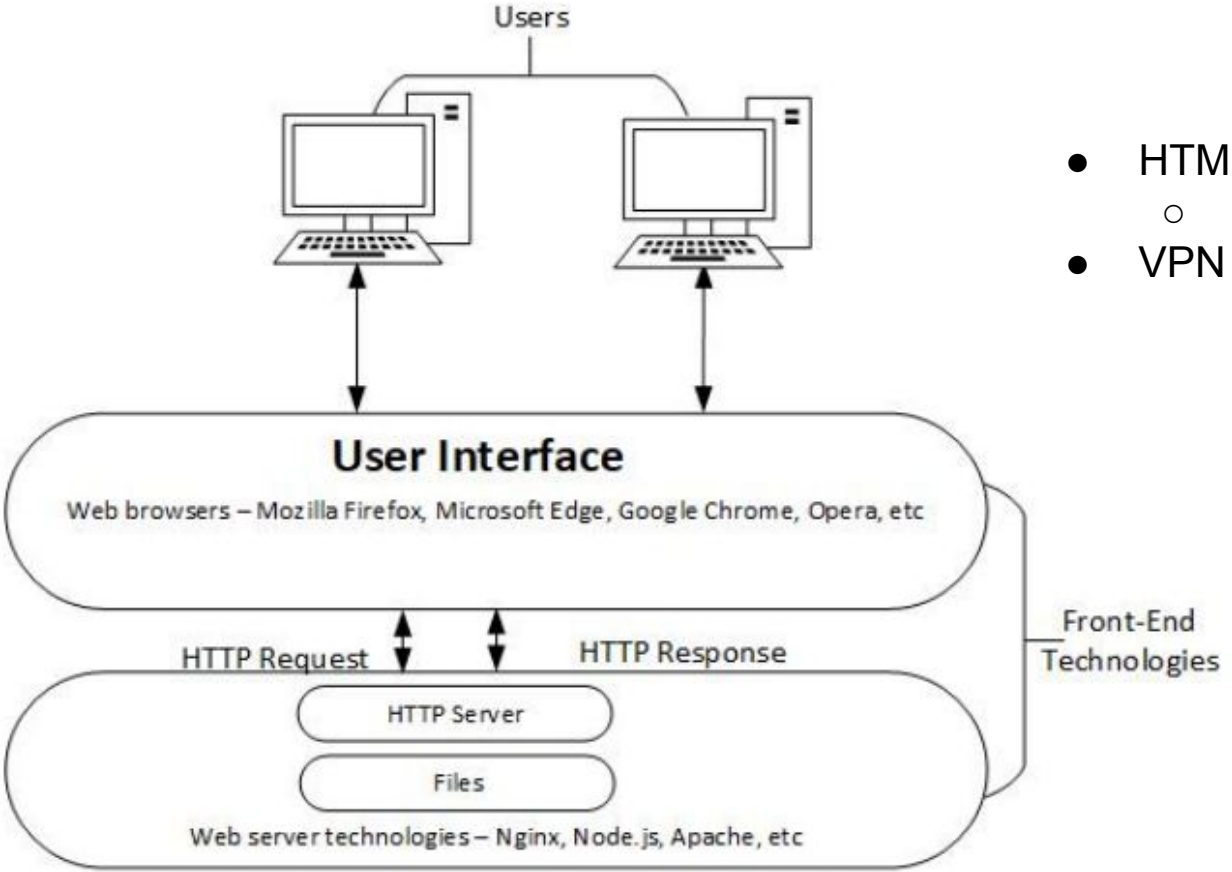
- Kibernetički poligoni su interaktivne *simulirane* reprezentacije organizacijskih mreža, sustava, alata i aplikacija (NIST 2018)
- Nude zaštićeno okruženje za legalno stjecanje kibernetičkih vještina, testiranje proizvoda i testiranje pripravnosti u organizaciji
- Vanjski sustavi su u pravilu simulirani (Internet, mediji, procesi, …)

Infrastruktura kiber. poligona - backend (Ukwandu 2020)



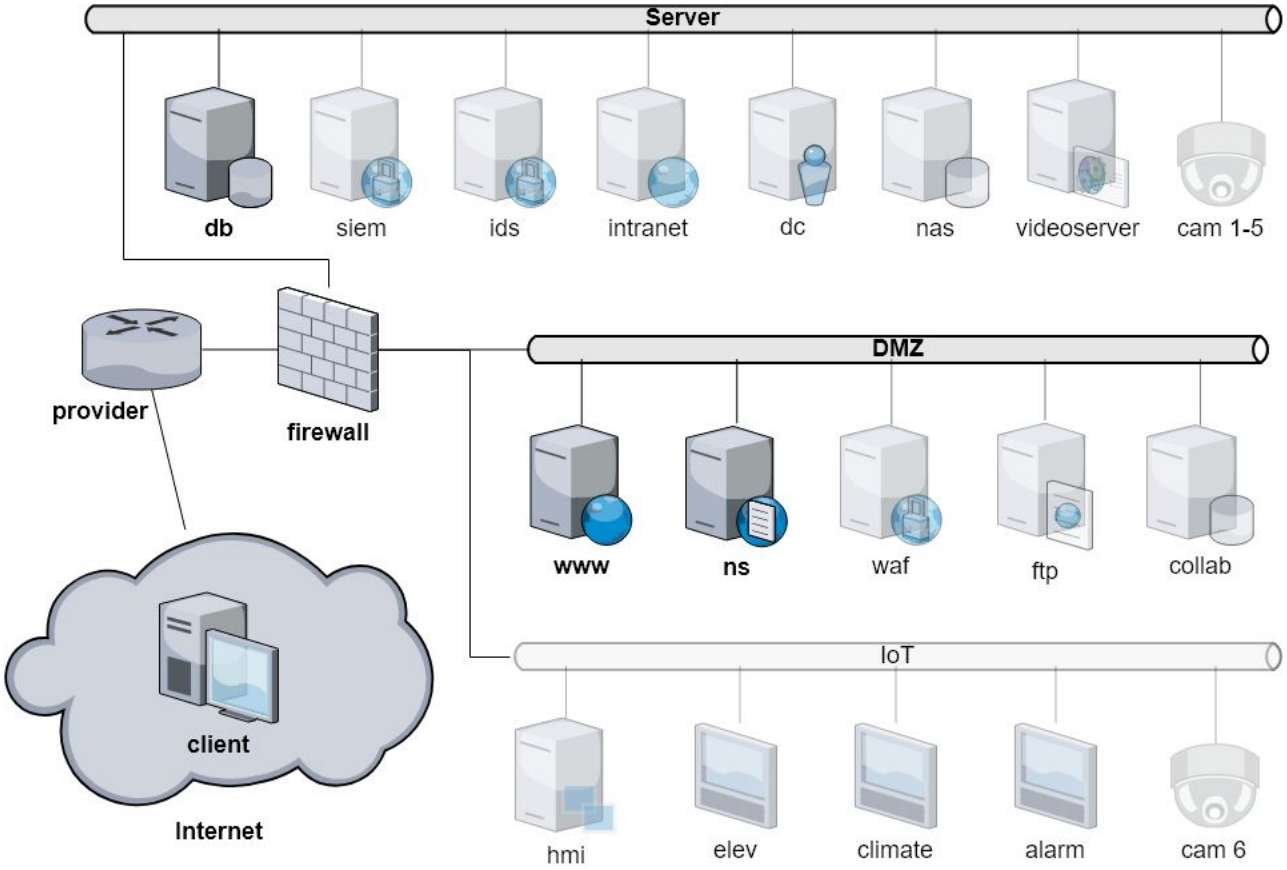
- Stvarne komponente, npr. PLC
- Primjer simulatora električne mreže: OPAL-RT

Infrastruktura kiber. poligona - frontend (Ukwandu 2020)



- HTML5 simulator terminala
  - npr. https://xtermjs.org
- VPN + udaljena ljuska

Primjer vježbenog okruženja (Russo 2020)



Uobičajeni koraci prilikom postavljanja poligona

1. Modeliranje vježbene infrastrukture (theatre)
2. Uvođenje ranjivosti za potporu vježbenog scenarija
3. Verifikacija modela i scenarija
4. Pokretanje vježbene infrastrukture
5. Testiranje vježbene infrastrukture

Obrađeni sustavi i pristupi

- NCR (Ferguson 2014)
- ALIVE (Braje 2016)
- CRATE (Gustafsson 2020)
- CRACK (Russo 2018, 2020)
- ADLES (de Leon 2018)
- ALPACA (Eckroth 2019)

NCR (Ferguson 2014)

- *National Cyber Range* - Ministarstvo obrane SAD-a
- Infrastruktura kibernetičkog poligona
- Prvenstveno namijenjen testiranju opreme i rješenja za vojne misije
- Integrira se u postojeće kolaboracijske alate DoD-a

ALIVE (Braje 2016)

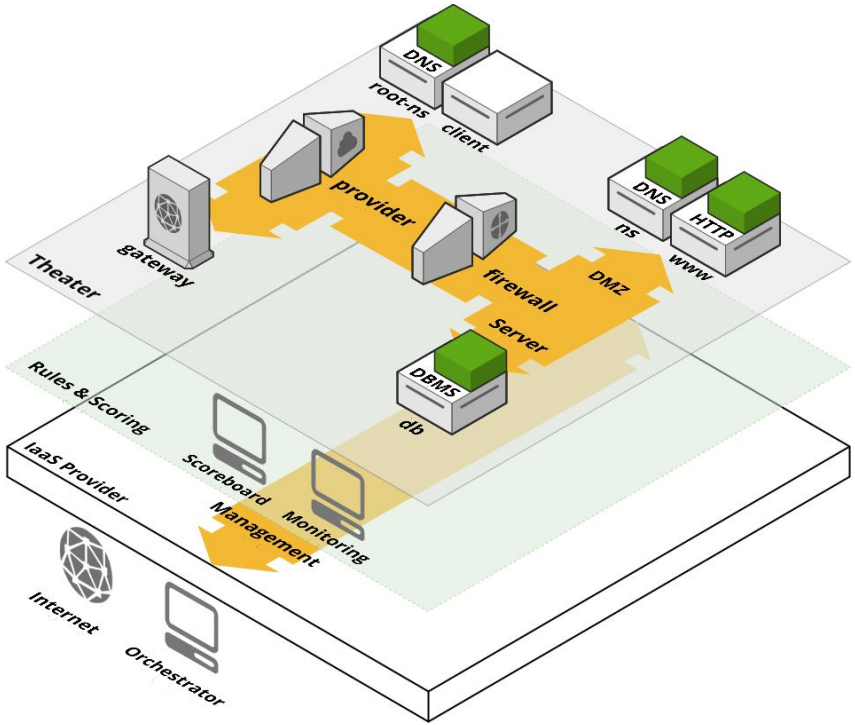
- *Automatic Live Instantiation of a Virtual Environment*
- Alati za potporu vježbi
- Razvija Lincoln Laboratory
- Glavni fokus je na generiranju realističnog prometa u poligonu

CRATE (Gustafsson 2020)

- Infrastruktura kibernetičkog poligona
- Održava ju Swedish defence research agency
- Potpora ICS vježbama: maketa grada s kritičnom infrastrukturom (>70 PLC-a)
- Daje potporu za uključivanje hardvera u vježbu

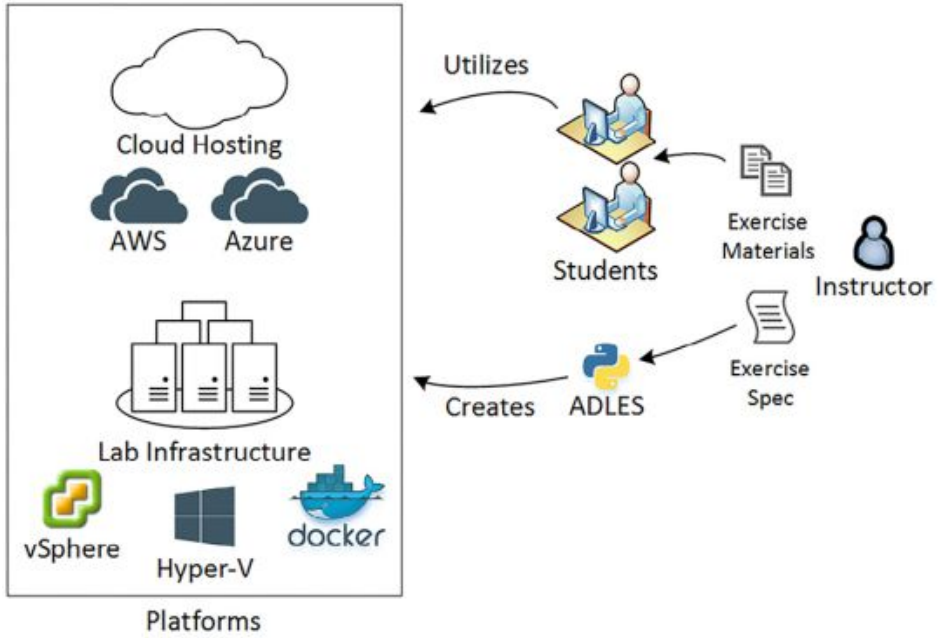
CRACK (Russo 2018, 2020)

- Alati za potporu razvoju vježbene okoline i scenarija
- Infrastruktura kibernetičkog poligona su vanjski davatelji IaaS-a
  - Cloudify, ARIA TOSCA, OpenTOSCA, Alien4Cloud, Heat-Translator
- Naglasak na interaktivnoj validaciji i testiranju modela poligona i scenarija
- https://github.com/enricorusso/CRACK



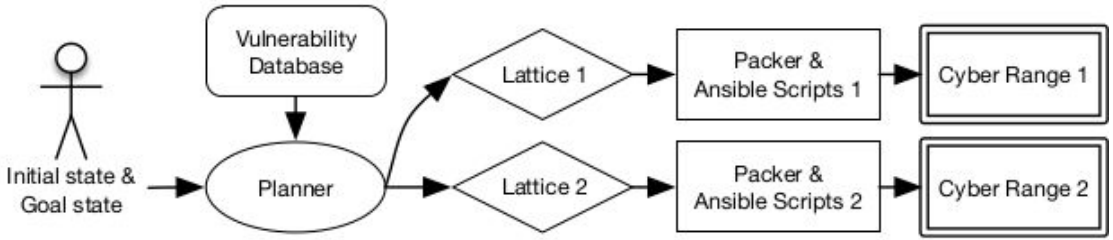
ADLES (de Leon 2018)

- *Automated Deployment of Lab Environments System*
- Alati za potporu vježbi koji koriste vanjske davatelje IaaS-a
- Fokusira se na podučavanje studenata i uključivanje nastavnih materijala u vježbu
- https://github.com/GhostofGoes/ADLES



ALPACA (Eckroth 2019)

- Alat za automatsko predlaganje vježbenih scenarija
- Trenutno stvara poligone sa samo jednim računalom
- https://github.com/StetsonMathCS/alpaca



Automatizacija na primjeru sustava CRACK (1)

- Vježba se modelira u jeziku CRACK SDL
- Model uključuje vježbenu infrastrukturu, ranjivosti i ciljeve napadača
- Ciljevi napadača i iskorištavanja ranjivosti su opisani u Datalog-u

```
www:
 type: Server
 properties:
  image: ubuntu1604
  flavor: medium
  requirements:
   - port: www_DMZ_port
www_DMZ_port:
 type: Port
 properties:
  fixed_ip: 198.51.100.5
 requirements:
  - network: DMZ
  - subnet: DMZ_subnet
```

```
DMZ:
 type: Network
DMZ_subnet:
 type: Subnet
 properties:
  subnet:
   cidr: 198.51.100.0/24
 requirements:
  - network: DMZ
```

Elevacija privilegija:

```
hasAccount(A, H, =>ToUser) :- hasUser(=>ToUser, H, P1, R1),
                              hasUser(=>FromUser, H, P2, R2),
                              hasAccount(A, H, =>FromUser).
```

Opis cilja: hasAccount(eve, www_system, root)?

Automatizacija na primjeru sustava CRACK (2)

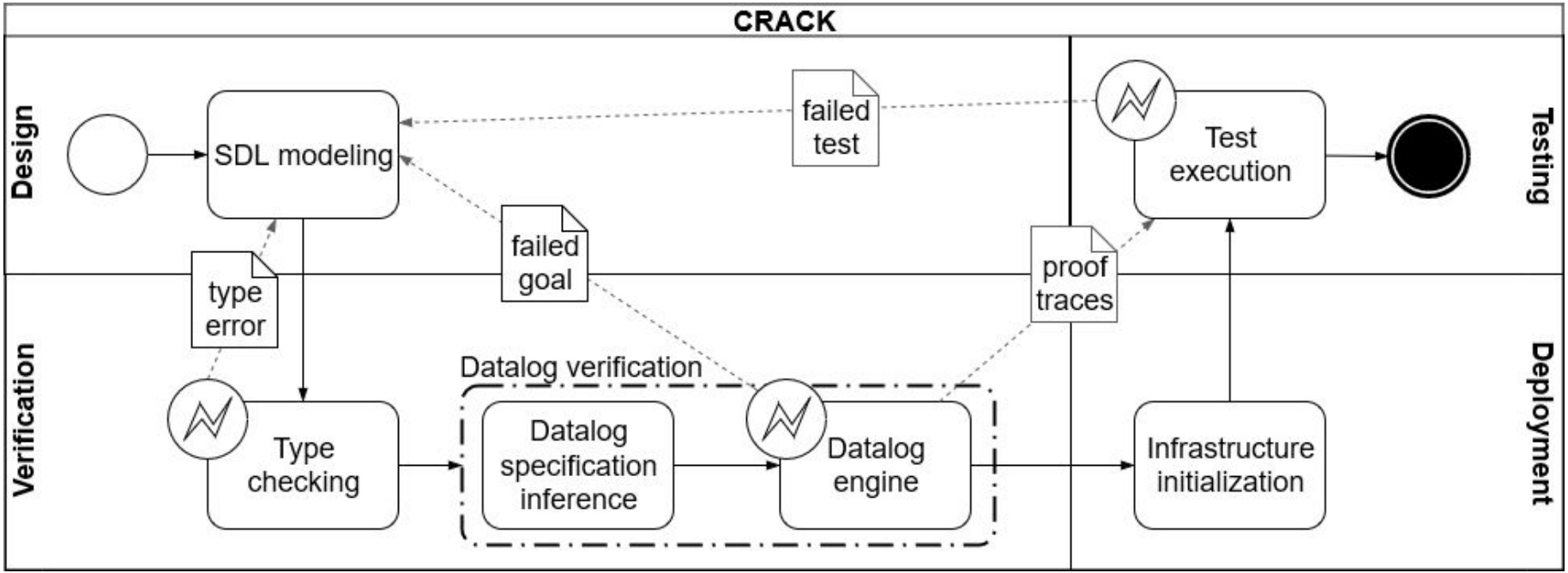
- Dobiveni model se verificira po pravilima pomoću Dataloga
  - Jesu li komponente ispravno opisane?
  - Jesu li postavljeni ciljevi napadača ostvarivi?
- Verifikacija vraća povratne informacije o greškama u modelu
- Na temelju modela sustav šalje upute davatelju IaaS-a i ondje postavlja infrastrukturu

Automatizacija na primjeru sustava CRACK (3)

- Konačni postavljeni poligon ne mora nužno odgovarati modelu
- CRACK na temelju Datalog tracea izrađuje testove
- Testovi se izvršavaju nad postavljenim poligonom
- Ako postoje odstupanja između modela i poligona, sustav prijavljuje pogreške

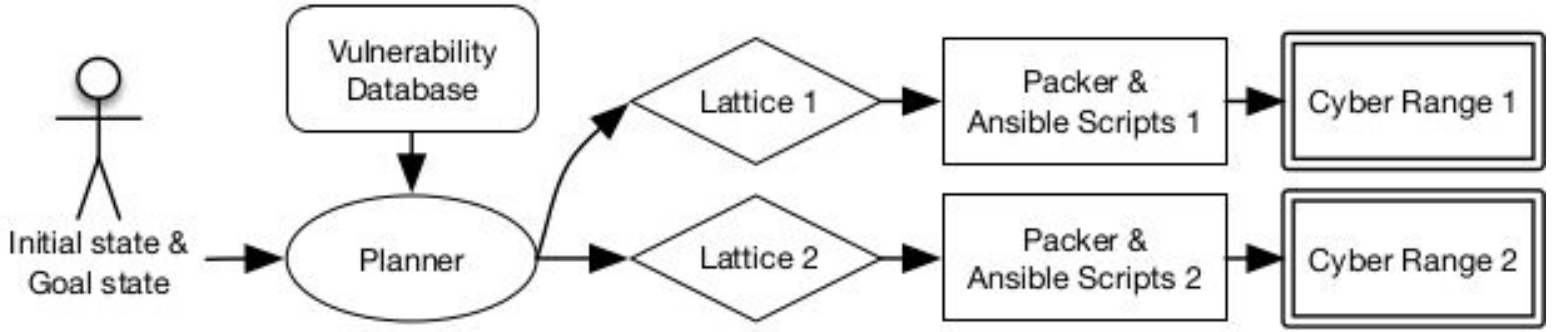
Automatizacija na primjeru sustava CRACK (4)

- Pregled cijelog postupka:



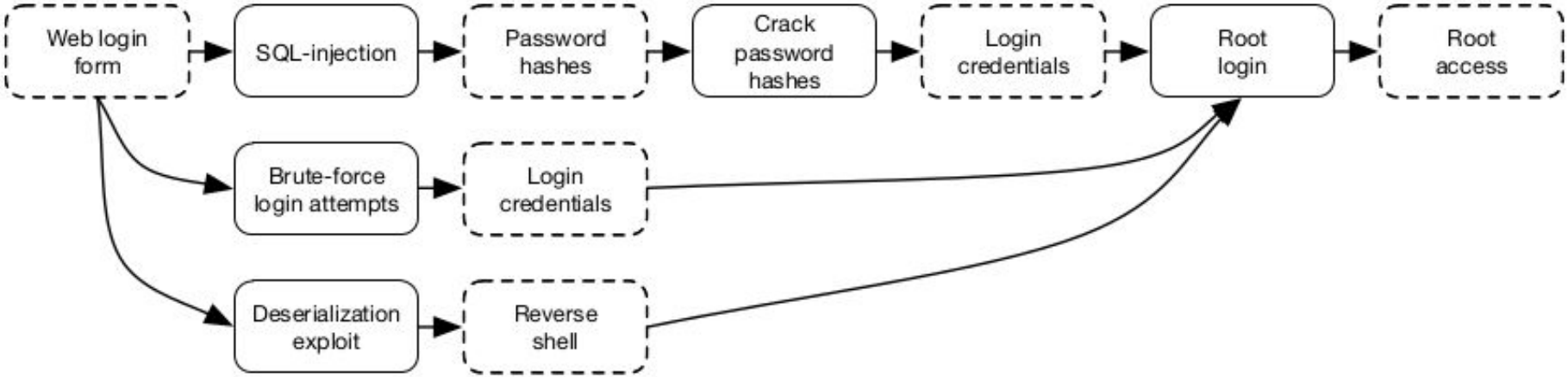
Izgradnja scenarija na primjeru sustava ALPACA (1)

- Ideja: Umjesto da se ranjivosti definiraju ručno, zadaje se početno i ciljno stanje sustava u vježbi (Prolog izrazi)
- Ranjivosti su opisane u bazi podataka - trenutno svega 19
    - Preduvjeti, posljedice, skripta za ugradnju
- Sustav automatski nalazi kombinacije ranjivosti koje vode do cilja



Izgradnja scenarija na primjeru sustava ALPACA (2)

- Autor može zadati željenu težinu scenarija



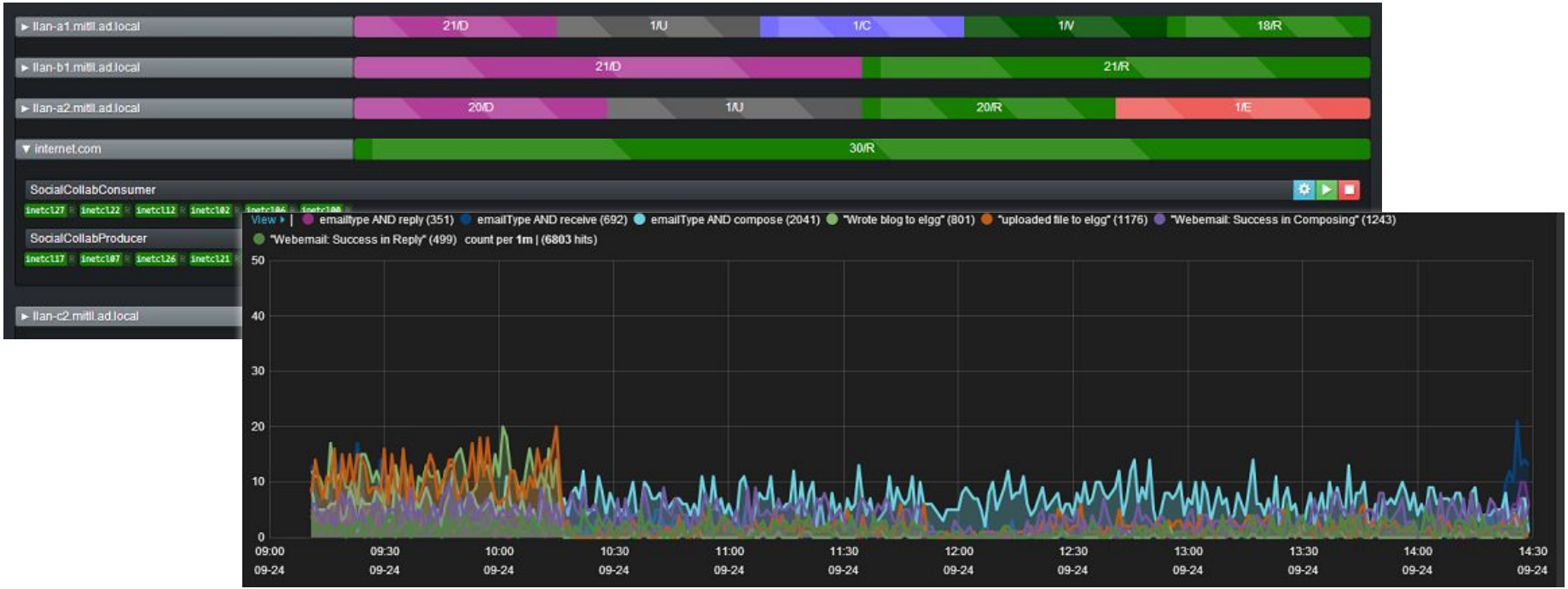
- Nije testirano na većem broju ranjivosti, pa je upitna skalabilnost
- Ne podržava mreže (radi sa samo jednim VM-om)

Sustav ALIVE (1)

- Sama potpora razvoju scenarija je usporediva s ostalim sustavima
- Uvodi domenski jezik za opis zaposlenika i poslovanja organizacije
- Modelira *realistično* ponašanje zaposlenika
  - Slanje i primanje e-pošte, pregledavanje web stranica, pisanje blogova, …
- Misije definiraju što grupe zaposlenika rade - ugrubo odgovaraju poslovnim procesima
- Simulirani zaposlenici su sami sposobni za fallback - ako ne radi mail klijent, pokušat će čitati mail putem webmail sučelja
- Rezultati: generiranje realističnog prometa, uvođenje dodatnih ranjivosti

Sustav ALIVE (2)

- Korisničko sučelje za praćenje rada simuliranih zaposlenika



Zaključci

- Postoje javno dostupni alati koji na temelju zadanog modela mogu automatski postaviti kibernetički poligon
- ALPACA podržava vrlo mali broj napada i samo jedan VM
- Modeliranje zaposlenika i generiranje prometa značajno ovise o podršci od strane infrastrukture kibernetičkog poligona (npr. NCR, ALIVE, CRATE) koja nije javno dostupna
- Izrada modela se i dalje obavlja ručno, uz automatsku verifikaciju

Reference

- Braje, Timothy M. Advanced tools for cyber ranges. MIT Lincoln Laboratory Lexington United States, 2016.
- Eckroth, Joshua, et al. "Alpaca: Building dynamic cyber ranges with procedurally-generated vulnerability lattices." Proceedings of the 2019 ACM Southeast Conference. 2019.
- Ferguson, Bernard, Anne Tall, and Denise Olsen. "National cyber range overview." 2014 IEEE Military Communications Conference. IEEE, 2014.
- Gustafsson, Tommy, and Jonas Almroth. "Cyber range automation overview with a case study of CRATE." Nordic Conference on Secure IT Systems. Springer, Cham, 2020.
- de Leon, Daniel Conte, et al. "ADLES: Specifying, deploying, and sharing hands-on cyber-exercises." Computers & Security 74 (2018): 12-40.
- Russo, Enrico, Gabriele Costa, and Alessandro Armando. "Scenario design and validation for next generation cyber ranges." 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). IEEE, 2018.
- Russo, Enrico, Gabriele Costa, and Alessandro Armando. "Building next generation cyber ranges with CRACK." Computers & Security 95 (2020): 101837.
- Ukwandu, Elochukwu, et al. "A review of cyber-ranges and test-beds: Current and future trends." Sensors 20.24 (2020): 7148.
- NIST. Cyber Ranges. https://www.nist.gov/system/files/ documents/2018/02/13/cyber_ranges.pdf; 2018. (Accessed on September 2021).

Q&A