Ekspertski sistem za detekciju anomalija u mrežnom saobraćaju i otkrivanje ranjivosti

Autori: Katarona Krstin SV57/2021
Jovan Vučković SV64/2021

Opis problema

Motivacija

Digitalne mreže su izložene stalnim pretnjama: pogrešno konfigurisani servisi, zastarele verzije softvera, slabosti u topologiji, kao i anomalije u saobraćaju (port scan, DDoS, beaconing). Tipični IDS/IPS alati zahtevaju kompleksnu tunning-konfiguraciju i ne nude lako prilagodljive ekspertske preporuke. **Ekspertski sistem zasnovan na pravilima (Drools)** omogućava transparentno i objašnjivo donošenje odluka: pravila su čitljiva i lako proširiva.

Pregled Problema

Postoje komercijalni sistemi (npr. Snort, Suricata), ali su rigidni, zahtevaju stalno ažuriranje i ne pružaju jednostavan način za prilagođavanje pravila u skladu sa poslovnim potrebama. Naš sistem će omogućiti **dinamičko donošenje odluka na osnovu pravila**, fleksibilnu nadogradnju baze znanja i integraciju sa događajima (CEP).

Predloženi sistem donosi:

- **Bazu znanja** o slabostima (portovi, servisi, verzije, konfiguracije).
- **CEP** nad mrežnim događajima (broj konekcija, pokušaji konekcija ka portovima, volumetrija) sa vremenskim prozorima.
- **Forward chaining** za derivaciju stanja (npr. „Uređaj je nesiguran” ⇒ „Uputi preporuku”), **backward chaining** za ciljno zaključivanje (npr. „Da li je mreža usaglašena?”).
- **Template** (DRT) za masovno generisanje pravila o portovima/servisima bez dupliciranja koda.

Metodologija rada

Ulaz u sistem (Input)

- **Lista aktivnih uređaja** i **IP adresa** u mreži
- **Lista otvorenih portova** i aktivnih servisa po uređajima
- **Logovi** mrežnog saobraćaja (broj konekcija, pokušaji konekcija na zabranjene portove, neobični paketi)
- Podaci o **poznatim ranjivostima** i preporukama
- **Mrežni događaji** (eventovi) u realnom vremenu:
    - `FlowEvent` (srcIP, dstIP, dstPort, proto, bytes, timestamp)
    - `AuthEvent` (ip, outcome, timestamp)
    - `DnsEvent` (ip, fqdn, timestamp)

Izlaz iz sistema (Output)

- **Detektovane ranjivosti** po uređajima i servisima (npr. Telnet na 23, SMBv1 zastareo, HTTP bez TLS-a).
- **Preporuke/akcije** (npr. „Zatvori port 23”, „Ažuriraj OpenSSH ≥ 8.9”, „Prebaci HTTP na HTTPS”).
- **Upozorenja na anomalije** (port scan, moguće DDoS, sumnjivo beaconing ponašanje).
- **Sumarni izveštaji** (broj nesigurnih uređaja, trendovi).

Baza znanja

Model domena (primer klasa/činjenica):

- `Device(id, ip, role, os, isIoT, hasFirewall)`
- `Service(deviceId, port, name, version)`
- `Vulnerability(deviceId, code, severity, description)`
- `Recommendation(deviceId, action, rationale)`
- `Alert(code, severity, context)`
- Event tipovi: `FlowEvent`, `AuthEvent`, `DnsEvent` (u `stream` sesiji za CEP)

Popunjavanje baze znanja:

Parsiranje nmap/asset skenera (ili generisani dataset)
→ insert `Device` i `Service` činjenica (uređaji, portovi, servisi, verzije).

Konfigurabilne liste i standardi bezbednosti:

- Lista „nesigurnih portova” (npr. 21 FTP, 23 Telnet, 445 SMB) → koristi se **DRL template** za pravila.
- Lista „minimalnih verzija servisa” (npr. SSH ≥ 8.9, SMB ≥ 3.0) → poređenje verzija za kreiranje ranjivosti.

Korišćenje CVE baze (cvedetails.com):

- Ako se na mreži pronađe servis određene verzije → proveri da li za tu verziju postoji poznata slabost (CVE).
- Generiši pravilo: *„Servis X verzija Y ima CVE slabost Z – preporučuje se update.”*

Mrežni agent ili simulacija šalje *Event*:

- `FlowEvent` za mrežni saobraćaj
- `AuthEvent` za prijave (uspeli/neuspeli login),
- `DnsEvent` za DNS upite.
  → ubacuje se u `KieSession` sa **@role(event)** i vremenskim prozorima (CEP).

Pravila za standardne portove:

- Ako se detektuje SSH ili Telnet otvoren na uređaju koji ne bi trebalo da ga ima (npr. desktop) → generiši preporuku za zatvaranje.

Pravila za pokušaje pristupa nezaštićenim servisima:

- Ako servis nije zahtevao autentikaciju ili TLS → označi kao nesiguran i preporuči zaštitu.

Analiza protoka saobraćaja (CEP):

- Ako se naglo poveća broj konekcija → mogući pokušaji **port scan** ili **DDoS**.
- Ako se detektuje veliki izlazni transfer (exfiltracija) → označi kao potencijalni data breach.

Uvid u izlazak podataka (data exfiltration):

- Ako se fajlovi koji ne bi smeli napustiti mrežu masovno šalju ka spoljnim IP-ovima → podigni alert **POTENTIAL_DATA_LEAK**.

Digitalna forenzika (dodatne opcije):

- **Pregled logova pristupa i autentikacije**:
  - Analiza abnormalnog broja neuspešnih login pokušaja (brute force).

    - Otkrivanje „lateral movement” – jedan nalog se prijavljuje na više uređaja u kratkom vremenu.
- **Detekcija malicioznih fajlova ili hash vrednosti**:
    - Uporedi hash fajlova sa poznatim IOC (Indicators of Compromise).
- **Korelacija događaja**:
    - Ako isti IP prvo radi port scan, a zatim šalje masivne FTP transfere → incident označen kao **COMPROMISED_DEVICE**.
- **Logovi procesa na hostovima**:
    - Ako se pokreću sumnjivi procesi (npr. `powershell` sa base64 komandama) → označi kao maliciozno.
- **Email forenzika (opciono)**:
    - Analiza zaglavlja i linkova u email porukama (phishing indikatori).

Primeri rezonovanja

Forward chaining

Primer A: Nesigurni servisi i grupni alarm

Cilj: Automatski označiti nesigurne uređaje i generisati preporuke; ako ih je mnogo, podići alarm.

Pravilo F1 — Telnet nesiguran:

```
rule "F1: Telnet insecure"
when
  $s : Service( name == "telnet" || port == 23, $dId : deviceId )
  $d : Device( id == $dId )
then
  insert( new Vulnerability($dId, "TELNET_INSECURE", "HIGH", "Telnet is insecure; disable.") );
end
```

Pravilo F2 — Preporuka gašenja usluge:

```
rule "F2: Recommend close port"
when
  $v : Vulnerability( code == "TELNET_INSECURE", $devId : deviceId )
then
  insert( new Recommendation($devId, "Close port 23 / remove telnet", "Insecure service") );
end
```

Pravilo F3 — Grupni alarm uz accumulate (≥3 uređaja):

```
rule "F3: Raise network alarm when many telnet vulns"
when
  Number( intValue >= 3 ) from accumulate (
    Vulnerability( code == "TELNET_INSECURE" ),
    count(1)
  )
then
  insert( new Alert("ALARM_TELNET_WIDESPREAD", "CRITICAL", "3+ devices with
Telnet") );
end
```

Primer B: Izračun rizika iz kombinacija

Cilj: Kombinovati više slabosti u „score” i odlučiti o jačoj akciji.

Pravilo F4 — HTTP bez TLS:

```
rule "F4: HTTP without TLS"
when
  $s : Service( name == "http", port == 80, $dId : deviceId )
then
  insert( new Vulnerability($dId, "HTTP_NO_TLS", "MEDIUM", "Serve over HTTPS")
);
end
```

Pravilo F5 — Zastarela SSH verzija:

```
rule "F5: Outdated SSH"
when
  $s : Service( name == "ssh", version < "8.9", $dId : deviceId )
then
  insert( new Vulnerability($dId, "SSH_OLD", "HIGH", "Upgrade OpenSSH >= 8.9") );
end
```

Pravilo F6 — Accumulate score i preporuka izolacije:

```
rule "F6: Risk score and isolate"
when
  $d : Device( $id : id )
  $score : Number( intValue >= 5 ) from accumulate(
     Vulnerability( deviceId == $id, $sev : severity ),
     sum( $sev == "HIGH" ? 3 : 2 )
  )
then
```

```
  insert( new Recommendation($id, "Isolate device from external network", "Risk
score >= 5") );
end
```

Backward chaining

Primer C: Usaglašenost mreže

Cilj: Odgovoriti na pitanje „Da li je mreža usaglašena?” (nema kritičnih slabosti).

Upit (query):

```
query "qCriticalVulnForDevice"( String $devId )
  Vulnerability( deviceId == $devId, severity == "HIGH" )
end
```

Pravilo B1 — Traži uređaj sa kritičnom slabošću (negacija):

```
rule "B1: Network not compliant if any HIGH"
when
  $d : Device( $id : id )
  exists( Vulnerability( deviceId == $id, severity == "HIGH" ) )
then
  insert( new Alert("NET_NOT_COMPLIANT", "HIGH", "At least one HIGH vuln") );
end
```

Pravilo B2 — Ako nema HIGH za sve uređaje → mreža usaglašena:

```
rule "B2: Network compliant"
salience -10
when
  not( Vulnerability( severity == "HIGH" ) )
then
  insert( new Alert("NET_COMPLIANT", "INFO", "No HIGH vulns present") );
end
```

Pravilo B3 — Na zahtev: dokaži uslov per-device (ciljno pozivanje upita):

```
rule "B3: Explain compliance per device"
when
  $d : Device( $id : id )
  not( qCriticalVulnForDevice( $id; ) )
then
  // explanation fact, npr. Evidence
end
```

Nivoi ulančavanja: B1/B2 + B3 (3 pravila u ciljno-vođenoj proceni usaglašenosti).

Primer D: Da li treba izolovati uređaj X?

Cilj: Dokaži predikat „`shouldIsolate(deviceId)`”.

Upit i pomoćna pravila:

```
declare Decision
  deviceId : String
  name     : String
end

query "qHighRisk"( String $devId )
  Number( intValue >= 5 ) from accumulate(
     Vulnerability( deviceId == $devId, $sev : severity ),
     sum( $sev == "HIGH" ? 3 : 2 )
  )
end

rule "B4: Should isolate if high risk or CEP alert"
when
  $d : Device( $id : id )
  ( qHighRisk( $id; ) or Alert( code == "CEP_SUS_BEACONING", context matches
$id ) )
then
  insert( new Decision($id, "ISOLATE") );
end

rule "B5: Should not isolate if critical service"
salience 5
when
  $d : Device( $id : id, role == "production-db" )
then
  insert( new Decision($id, "KEEP_ONLINE") );
end
```

CEP

Deklaracije događaja:

```
declare FlowEvent
  @role( event )
  @timestamp( timestamp )
  srcIP : String
```

```
  dstIP : String
  dstPort : int
  bytes : long
  timestamp : java.util.Date
end
```

C1 — Port scanning (više od 20 različitih portova u 10s sa iste IP):

```
rule "C1: Port scan"
when
  $src : String() from accumulate(
    FlowEvent( $s : srcIP ) over window:time(10s),
    collectSet( $s )
  )
  Number( intValue >= 20 ) from accumulate(
    FlowEvent( srcIP == $src ) over window:time(10s),
    countDistinct( dstPort )
  )
then
  insert( new Alert("CEP_PORT_SCAN", "HIGH", $src) );
end
```

C2 — Mogući DDoS (≥1000 konekcija ka istom odredištu u 5s):

```
rule "C2: DDoS suspect"
when
  $dst : String() from accumulate(
    FlowEvent( $d : dstIP ) over window:time(5s),
    collectSet( $d )
  )
  Number( intValue >= 1000 ) from accumulate(
    FlowEvent( dstIP == $dst ) over window:time(5s),
    count(1)
  )
then
  insert( new Alert("CEP_DDOS", "CRITICAL", $dst) );
end
```

C3 — Beaconing (periodični mali tokovi ka istoj destinaciji):

```
rule "C3: Beaconing pattern"
when
  Number( intValue >= 5 ) from accumulate(
    FlowEvent( bytes < 200 ) over window:time(2m),
    count(1)
```

```
  )
then
  insert( new Alert("CEP_SUS_BEACONING", "HIGH", "Frequent small flows") );
end
```

Template

device-service-vuln.drt (skraćeno):

```
template header
port
serviceName
minVersion
code
severity
message
recommendation

package rules.vuln

rule "T_${code}_${port}"
when
  $s : Service( port == @{port}, name == "@{serviceName}", version <
"@{minVersion}", $dId : deviceId )
then
  insert( new Vulnerability($dId, "@{code}", "@{severity}", "@{message}") );
  insert( new Recommendation($dId, "@{recommendation}", "Triggered by template")
);
end
```

CSV (primer):

```
port,serviceName,minVersion,code,severity,message,recommendation
21,ftp,1.0,FTP_OLD,MEDIUM,Outdated FTP server,Upgrade or disable FTP
25,smtp,2.0,SMTP_OLD,MEDIUM,Outdated SMTP server,Upgrade SMTP or use
relay with TLS
139,smb,2.1,SMBv1, HIGH,SMBv1 is insecure,Disable SMBv1 / upgrade
```