

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 tuning-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, Warnings)
- **Warning**(code, context, severity)
- 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 \geq 8.9, SMB \geq 3.0) → poređenje verzija za kreiranje ranjivosti.

Korišćenje CVE baze (cvedetails.com):

- Ako se na mreži pronađe servis određene verzije → proveriti 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 preporučiti 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.²

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
```

Primer C: Brute force Login Detection

Cilj: Blokirati IP adresu sa koje dolazi napad.

Pravilo F7 — Detekcija pokušaja napada:

```
rule "Detect multiple failed logins"
when
    $ip : String() from accumulate(
        $e : LoginEvent(success == false, $ip : ip) over window:time(5m),
        count($e) >= 5
    )
then
    // Kreiramo privremeni flag da označimo detekciju brute force patterna
    insertLogical("BRUTE_FORCE_DETECTED_" + $ip);
    output.add("[F1] Brute force detected for IP: " + $ip);
    System.out.println("[RULE F1] Brute force detected for IP: " + $ip);
end
```

Pravilo F8 — Postavljanje IP adrese na sumnjivu:

```
rule "Mark IP as suspicious"
when
    $flag : String( this.startsWith("BRUTE_FORCE_DETECTED_"), $ip : substring(21)
)
then
    insert(new SuspiciousIP($ip));
    output.add("[F2] Suspicious IP added: " + $ip);
    System.out.println("[RULE F2] Suspicious IP: " + $ip);
end
```

Pravilo F9 — Blokiranje sumnjive IP adrese:

```
rule "Block suspicious IP"
when
    $sus : SuspiciousIP($ip : ip)
then
    insert(new BlockAction($ip));
    output.add("[F3] Block action for IP: " + $ip);
    System.out.println("[RULE F3] Block action triggered for IP: " + $ip);
end
```

end

then

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

end

Primer D: Data Exfiltration

Cilj: Blokirati saobraćaj ukoliko se primeti kradja podataka.

Pravilo F10 — Suspicious Download Pattern

- Višestruki preuzeti fajlovi sa kritičnih servera u kratkom vremenu.
- Kreira SuspiciousDownloadActivity.
- Svrha: detektuje prvi znak potencijalnog insider napada.

rule "Detect suspicious download activity"

when

```
$user : String() from accumulate(  
    DownloadEvent($u : user) over window:time(5m),  
    count(DownloadEvent()) >= 10  
)
```

then

```
insert(new SuspiciousDownloadActivity($user));  
output.add("[F1] Suspicious download activity detected for user: " + $user);  
System.out.println("[RULE F1] Suspicious download activity detected for user: " +  
$user);  
end
```

Pravilo F11 — Outbound Connection Detected

- Isti korisnik/host otvara outbound konekciju ka nepoznatoj eksternoj adresi.
- Kreira PotentialExfiltration.
- Svrha: ovo je drugi, nezavisan signal – sama aktivnost download-a nije dovoljna da pokrene alert, ali sada povezujemo sumnjivog korisnika sa mrežnom aktivnošću.

rule "Detect potential exfiltration"

when

```
$conn : OutboundConnectionEvent($user : user, destinationType ==  
"UNKNOWN")  
SuspiciousDownloadActivity(user == $user)
```

then

```

insert(new PotentialExfiltration($user));
output.add("[F2] Potential data exfiltration detected for user: " + $user);
System.out.println("[RULE F2] Potential data exfiltration detected for user: " +
$user);
end

```

Pravilo F12 — Escalation / Alert

- Ako postoji PotentialExfiltration, plus dodatni uslov (npr. drugi download ili više od 10 fajlova), kreira AlertAdminAction ili BlockTrafficAction.
- Svrha: treći korak agregira informacije iz F1 i F2, uvodi dodatni uslov, i tek tada pokreće akciju.

```

rule "Escalate and alert admin"
when
    $exfil : PotentialExfiltration($user : user)
    Number(intValue >= 15) from accumulate(
        DownloadEvent(user == $user) over window:time(5m),
        count(DownloadEvent())
    )
then
    insert(new AlertAdminAction($user));
    output.add("[F3] ALERT: Admin notified for confirmed exfiltration by user: " +
$user);
    System.out.println("[RULE F3] ALERT triggered for user: " + $user);
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

```

Backward chaining

Fajl sistem je stablo direktorijuma koje će biti skenirano u cilju pronalaženja virusa u njima.

- **Čvor stabla** predstavlja direktorijum.
- **Listovi stabla** predstavljaju fajlove unutar tog direktorijuma.
- Svaki fajl se proverava na prisustvo **binarnih fingerprinta virusa**.
- Ako je fajl kompromitovan, direktorijum u kojem se nalazi se takođe označava kao kompromitovan, i taj status se propagira unazad (prema root-u).

Ovakvo stablo može se koristiti za **backward chaining zaključivanje**: da bi se utvrdilo da li je root direktorijum bezbedan, sistem mora da proveriti sve njegove poddirektorijume i fajlove.

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
```