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”)
- **Template** (DRT) za masovno generisanje pravila o slabostima portova/servisa 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):
    - `PacketEvent` (sourceIP, destinationIP, sourcePort, destinationPort, protocol, flags, payloadSize, dnsQuery, executionTime)
    - `Protocol` (TCP, UDP, DNS, ICMP)
    - `Flag` (SYN, ACK, RST)

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).
- **Sumirani izveštaji** (broj nesigurnih uređaja, trendovi).
- **Detektovane zaražene datoteke**.

Baza znanja

Model domena (primer klasa/činjenica):

- `Device(id, ip, osName, osVersion, osType)`
- `NetworkService(id, device, port, name, version)`
- `Vulnerability(id, device, code, severity, description)`
- `Recommendation(id, action, rationale)`
- `Alert(id, code, severity, description, executionTime)`
- `Log(dateTime, type, sourceIp, description, logTag)`
- Event tipovi: `PacketEvent`, `Alert` (u `stream` sesiji za CEP)

Popunjavanje baze znanja:

Parsiranje nmap/asset skenera (ili generisani dataset)
→ insert `Device` i `NetworkService` 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.”*

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.

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.
- **Korelacija događaja**:
  - Ako isti IP prvo radi port scan, a zatim šalje masivne FTP transfere → incident označen kao **COMPROMISED_DEVICE**.

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 "F7: Detect multiple failed login attempts from same IP"
when
    $l : Log( type == LogType.INFO, $srcIP : sourceIP, description.contains("failed"))
    $count : Number( intValue >= 3 ) from accumulate (
        Log( type == LogType.INFO, sourceIP == $srcIP, description.contains("failed")),
        count($l.getSourceIP())
    )
then
    insert(new Alert("Suspicions IP", Severity.MID, "Suspicious IP - A lot of failed login
attempts: " + $l.getSourceIP()));
end
```

Pravilo F8 — Postavljanje IP adrese na sumnjivu:

```
rule "F8: Escalate alert for suspicious IP with malicious behavior"
when
    $log : Log(
        type == LogType.WARNING,
        logTag == LogTag.ACCESS || logTag == LogTag.ACTION,
        $ip : sourceIP,
        description.toLowerCase matches
"(.*private.*|.*restricted.*|.*sensitive.*|.*attack.*|.*exploit.*)"
    )

    $alert : Alert(
        severity == Severity.MID,
        code == "Suspicions IP",
        description.contains($ip)
    )
then
    Alert highAlert = new Alert(
        "Malicious IP",
        Severity.HIGH,
        "IP address " + $ip + " is exhibiting malicious behavior: attempted access to
sensitive parts of the system."
    );
    insert(highAlert);
end
```

Pravilo F9 — Blokiranje sumnjive IP adrese:

```
rule "F9: Recommend blocking IP after multiple malicious alerts"
when
    $l : Log( type == LogType.WARNING, $ip : sourceIP, description.toLowerCase
matches "(.*private.*|.*restricted.*|.*sensitive.*|.*attack.*|.*exploit.*)")
    $count : Number( intValue >= 3 ) from accumulate (
        Alert(
            severity == Severity.HIGH,
            code == "Malicious IP",
            description.contains($ip)),
        count($l.getSourceIP())
    )
then
Recommendation rec = new Recommendation(
        "Block IP " + $ip,
        "Detected 3 or more malicious activities from IP " + $ip + ". Immediate blocking
is recommended."
    );
```

```
    insert(rec);
end
```

Primer D: SQL Injection

Cilj: Blokirati saobraćaj ukoliko se primeti pokušaj SQL injection-a.

Pravilo F10 — Suspicious Database Access

- Neuspeli query za bazu sa nepoznate IP adrese .
- Kreira Alert da se baza podataka ponaša čudno.
- Svrha: detektuje prvi znak potencijalnog insider napada.

```
rule "F10: Suspicious Database Access"
when

    $l : Log( type == LogType.INFO, logTag == LogTag.DATABASE, $ip : sourceIP,
description.contains("denied"))
        $count : Number( intValue >= 3 ) from accumulate (
            Log( type == LogType.INFO, sourceIP == $ip,
description.contains("denied")),
            count($l.getSourceIP())
        )

then
    insert(new Alert("Suspicious Database Access", Severity.MID, "Suspicious IP -
Multiple Accesses: " + $ip));
end
```

Pravilo F11 — SQL Injection detection

- Isti korisnik/host pravi sumnjivi query sa sql injection upitom.
- Kreiraj high risk Alert.

```
rule "F11: SQL Injection"
when
    // Postoji novi log koji je ozbiljniji — pokušaj pristupa osetljivim delovima
    $log : Log(
        type == LogType.WARNING,
        logTag == LogTag.ACCESS || logTag == LogTag.ACTION,
        $ip : sourceIP,
        description.toLowerCase matches ".*(\\b(union select|select .* from|insert
into|update .* set|delete
from)\\b|\\bor\\s+1\\s*=\\s*1\\b|\\b'\\s+or\\s+'1'\\s*=\\s*'1\\b|\\b\"\\s+or\\s+\"1\"\\s*=\\s+\"
```

```
1\"\\b|--|;--|/\\*|\\*/|\\bbenchmark\\s*\\(|\\bsleep\\s*\\(|\\binformation_schema\\b|\\bconc
at\\s*\\(|\\bxp_cmdshell\\b).*"
    )

    // Postoji već alert za sumnjiv IP sa Severity.MID
    $alert : Alert(
        severity == Severity.MID,
        code == "Suspicious Database Access",
        description.toLowerCase contains $ip
    )
then
    Alert highAlert = new Alert(
        "Malicious Database Action",
        Severity.HIGH,
        "IP address " + $ip + " is exhibiting malicious behavior: attempted sql injection."
    );
    insert(highAlert);
end
```

Pravilo F12 — Blocking IP after SQL Injection

- Ako postoji High riisk alert za sql injection vise od 1 za istu ip adresu ta ip adresa se bloki

```
rule "F12: Blocking IP after SQL Injection"
when

    $l : Log( type == LogType.WARNING, logTag == LogTag.ACCESS, $ip : sourceIP,
description.toLowerCase matches ".*(\\b(union select|select .* from|insert into|update
.* set|delete
from)\\b|\\bor\\s+1\\s*=\\s*1\\b|\\b'\\s+or\\s+'1'\\s*=\\s*'1\\b|\\b\"\\s+or\\s+\"1\"\\s*=\\s+\"
1\"\\b|--|;--|/\\*|\\*/|\\bbenchmark\\s*\\(|\\bsleep\\s*\\(|\\binformation_schema\\b|\\bconc
at\\s*\\(|\\bxp_cmdshell\\b).*")
        $count : Number( intValue >= 3 ) from accumulate (
            Alert(
                severity == Severity.HIGH,
                code == "Malicious Database Action",
                description.contains($ip)),
            count($l.getSourceIP())
        )

then
    Recommendation rec = new Recommendation(
        "Block IP " + $ip,
```

```
        "Detected  SQL Injection from IP " + $ip + ". Immediate blocking is
recommended."
    );
    insert(rec);
end
```

CEP

Postoje 2 vrste cep-a koje snimamo u našoj aplikaciji:

- Snimanje izlaznog saobeaćaja iz mreže (outbound)
- Snimanje ulaznog saobraćaja u mrežu (inbound)

Korišćenjem ove 2 vrste kompleksnih događaja uspešno detektujemo maliciozne pakete koji se kreću našim mrežnim saobraćajem. Podeljeni su u 2 .drl fajla. Stvari na koje se obraća posebna pažnja prilikom detekcije malicioznih paketa su: veličina payload-a, sumnjive IP adrese, sumnjivi patterni kretanja paketa.

Obe grupe pravila koriste zajednički model događaja `PacketEvent` i generišu događaje tipa `Alert` i `Recommendation`, koji se dalje prosleđuju analitičkom servisu i interfejsu za prikaz upozorenja.

Na ovaj način aplikacija omogućava detekciju različitih tipova mrežnih napada u realnom vremenu, i to kako na ulazu, tako i na izlazu iz mreže.

Template

Postoje 2 vrste template-a u našoj aplikaciji:

- Template za generisanje pravila na osnovu kojih se detektuju ranjivosti servisa - za generisanje ovih pravila korišćena je MITRE CVE baza znanja
- Template za generisanje pravila na osnovu kojih se detektuju ranjivosti vezane za operativni sistem - za generisanje ovih pravila korišćena je MITRE CVE baza znanja.

Svaki template sadrži parametre koji se popunjavaju na osnovu CVE podataka (npr. `CVE_ID`, `affected_component`, `severity`, `exploit_pattern`) i koristi se za kreiranje `.drl` pravila koja se automatski učitavaju u CEP engine.

Na ovaj način se postiže **automatizovana adaptacija sistema** na nove pretnje, bez potrebe za ručnim pisanjem svakog pravila.