

Informacione Tehnologije

# IoT Forensic Hub

*Hibridni sistem za autonomnu forenzičku analizu i detekciju  
botnet aktivnosti u kriptovanom saobraćaju primenom Isolation  
Forest modela*

**Predmet:** Internet Inteligentnih Uredaja

**Autor:** Nemanja Domanović

**Autor:** Nenad Bogosavljević

# Sadržaj

|  |           |
|--|-----------|
| <b>1 Uvod: Sigurnosne pretnje i evolucija</b>                          | <b>2</b>  |
| 1.1 Inženjerski izazov: Detekcija bez Payload-a . . . . .              | 2         |
| <b>2 Faza I: Forenzičko prepoznavanje u okruženju po imenu Malcolm</b> | <b>2</b>  |
| 2.1 Ekstrakcija Artefakata i Zeek Log Analiza . . . . .                | 2         |
| 2.2 Arkime Timeline: Vizuelizacija Botnet Pulsa . . . . .              | 3         |
| <b>3 Faza II: DSWVA — Algoritam</b>                                    | <b>4</b>  |
| <b>4 Faza III: Mašinsko Učenje i Detekcija Anomalija</b>               | <b>4</b>  |
| 4.1 Logika Izolacije (Isolation Forest) . . . . .                      | 4         |
| <b>5 Faza IV: UI korisnički interfejs</b>                              | <b>5</b>  |
| 5.1 Faza V: Validacija i Laboratorijska Evaluacija . . . . .           | 7         |
| 5.2 Laboratorijski mod: Ubacivanje i Generisanje Dokaza . . . . .      | 7         |
| 5.3 Analiza Preciznosti i Recall-a . . . . .                           | 7         |
| 5.4 Interpretacija rezultata: Optimalni status . . . . .               | 8         |
| 5.5 Faza VI: Analiza Mrežnog Sloja (Vector Analysis) . . . . .         | 9         |
| <b>6 NACRT PATENTNE PRIJAVE (Tehnički opis)</b>                        | <b>10</b> |
| 6.1 Definisanje problema . . . . .                                     | 10        |
| 6.2 Suština pronalaska . . . . .                                       | 10        |
| 6.3 Tehnički crteži (Arhitektura, Pipeline, Logika) . . . . .          | 10        |
| 6.3.1 Crtež 1: Opšta arhitektura integracije sistema . . . . .         | 10        |
| 6.3.2 Crtež 2: Pipeline logičkog procesiranja podataka . . . . .       | 11        |
| 6.3.3 Crtež 3: Sekvencijalni dijagram vremenske analize . . . . .      | 11        |
| 6.4 Definisanje inovativnog doprinosa kroz patentne zahteve . . . . .  | 12        |
| <b>7 Zaključak</b>   | <b>12</b> |

# 1 Uvod: Sigurnosne pretnje i evolucija

Eksponencijalni rast IoT (Internet of Things) uređaja doveo je do stvaranja velikog broja pretnji koje uglavnom nemaju značajne nivoe zaštite. Uređaji poput pametnih kamera, kućnih senzora i industrijskih rutera poseduju minimalne bezbednosne mehanizme, što ih čini idealnim metama za formiranje botnet mreža.

## 1.1 Inženjerski izazov: Detekcija bez Payload-a

Problem koji ovaj rad rešava je detekcija maliciozne komunikacije u uslovima kada je mrežni saobraćaj kriptovan (TLS/SSL). Pošto sadržaj paketa nije dostupan, IoT Forensic Hub koristi analizu temporalnih metapodataka (Inter-Arrival Time - IAT). Prepostavka rada je da malver, izvršavajući fiksne algoritme, ostavlja svojevrsni "otisak prsta" na mrežnom sloju koji je nemoguće sakriti bez degradacije efikasnosti samog napada.

# 2 Faza I: Forenzičko prepoznavanje u okruženju po imenu Malcolm

Analiza je započeta upotrebom **Malcolm** platforme, koja integriše Zeek IDS i Arki-me. Malkolm je moćan alat za analizu mrežnog saobraćaja otvorenog koda, dizajniran da poboljša bezbednosne operacije preduzeća. Razvijen od strane CISA u saradnji sa Nacionalnom laboratorijom Ajdaha (INL), Malkolm omogućava bezbednosnim timovima da obrađuju, obogaćuju i vizuelizuju mrežnu telemetriju za otkrivanje pretnji, usklađenost i forenziku. U našem radu, ovaj alat nam je poslužio za definisanje „normalnog“ i „malicioznog“ stanja sistema.

## 2.1 Ekstrakcija Artefakata i Zeek Log Analiza

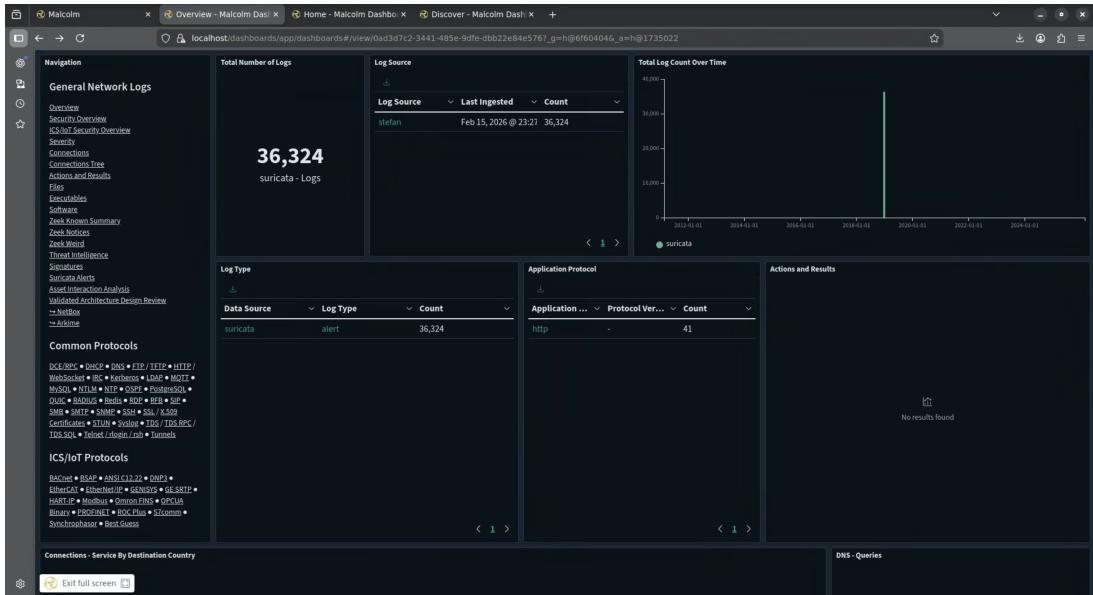
U ovoj fazi smo identifikovali da botnet sesije karakteriše ekstremno mala varijansa vremenskih razmaka.

```

1 def forensic_ingest(file_path):
2     df = pd.read_csv(file_path)
3     # Pretvaranje vremenskih pecata u nanosekundnu preciznost
4     df['ts'] = pd.to_datetime(df['ts'], unit='s')
5     df = df.sort_values(by='ts')
6     # Diferencijalna analiza vremena dolaska
7     df['delta'] = df['ts'].diff().dt.total_seconds().fillna(0)
8     return df

```

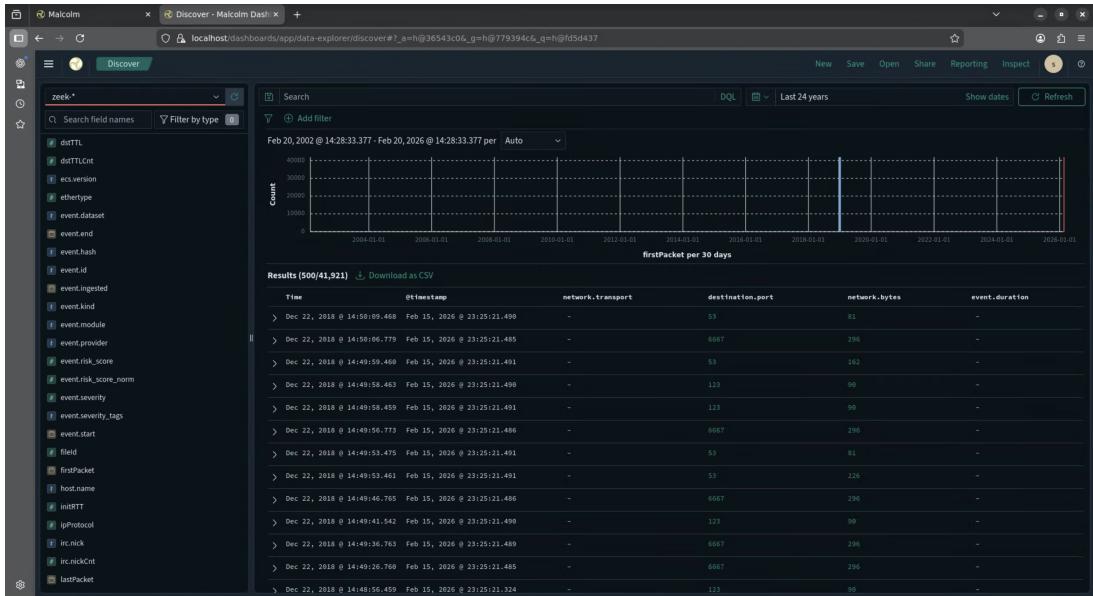
Prikaz koda 1: Ingestija i čišćenje mrežne telemetrije



Slika 1: Vizuelni pregled mrežnih sesija i protokola unutar Malcolm Dashboard-a.

## 2.2 Arkime Timeline: Vizuelizacija Botnet Pulsa

Vizuelna validacija u Arkime modulu otkrila je "Beaconing" proces — periodično slanje signalja ka C2 serveru koji se na timeline-u vidi kao savršena simetrična distribucija impulsa.



Slika 2: Prikaz precizne periodičnosti paketa zaraženog uređaja.

### 3 Faza II: DSWVA — Algoritam

Algoritam *Delta-Sub-millisecond Variance Analysis* (DSWVA) predstavlja u ovom primeru rešenje za prepoznavanje automatskog delovanja. Suština DSWVA algoritma je na identifikaciji diskretnih razlika između nasumične prirode ljudske interakcije i unapred definisanog, ponavljajućeg ponašanja automatskih procesa. Dok korisnici generišu mrežni saobraćaj sa visokim stepenom varijabilnosti, botnet malveri (poput Mirai ili Gafgyt) izvršavaju svoje instrukcije unutar programskih petlji.

Ova automatizacija se prikazuje kao serija mrežnih paketa čija vremenska odstupanja (IAT - Inter-arrival Time) teže nuli u idealnim uslovima. DSWVA to beleži merenjem varijanse drugog reda, čime izoluje stabilne bot interakcije od nasumičnog šuma regularnog saobraćaja. Na ovaj način, sistem se stvara pojam ponašanja bot saobraćaja – **Indeks automatizacije**.

```

1 def dswva_score(df):
2     # Analiza stabilnosti varijanse u klizecm prozoru
3     df['iat_var'] = df['delta'].rolling(window=50).var()
4     # Ukupna devijacija od determinističkog modela
5     total_var = df['delta'].var()
6     # Formula za Automation Index
7     score = 100 * (1 / (1 + total_var * 1000))
8     return score

```

Prikaz koda 2: DSWVA kalkulacija varijanse drugog reda

### 4 Faza III: Mašinsko Učenje i Detekcija Anomalija

Najvažnija tačka ovog rada je upotreba nenadgledanog (unsupervised) modela za detekciju pretnji koje namerno unose mrežni šum (jitter).

#### 4.1 Logika Izolacije (Isolation Forest)

Za razliku od klasičnih ML modela, **Isolation Forest** ne traži sličnost, već izoluje anomalije. Anomalije su retke i bitno se razlikuju u prostoru obeležja.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \quad (1)$$

Model gradi skup stabala, gde su napadi (anomalije) laksi za izolaciju jer imaju kraće putanje do korena stabla.

```

1 from sklearn.ensemble import IsolationForest
2
3 def execute_ai_analysis(df):
4     # n_estimators=150: Povećava stabilnost particionisanja
5     model = IsolationForest(n_estimators=150, contamination=0.01)
6     # Fit i predikcija (Labela -1 označava pretnju)
7     df['threat_id'] = model.fit_predict(df[['delta', 'iat_var']])
8
9 \end{threat_triaing}

```

Prikaz koda 3: Implementacija Isolation Forest modela

## 5 Faza IV: UI korisnički interfejs

Korisnički interfejs je dizajniran da simulira mrežni operativni centar (SOC).



Slika 3: Glavni dashboard sistema sa glavnim indikatorima u realnom vremenu.

Hronološka analiza (Timeline analysis) predstavlja metod rekonstrukcije mrežnih događaja u hronološkom redosledu, čime se omogućava uvid u tačan sled aktivnosti unutar određenog vremenskog okvira. U kontekstu ovog rada, alat je ključan za vizuelnu identifikaciju beaconing-a, jer jasno razotkriva strogu periodičnost robotske komunikacije koju je nemoguće uočiti prostim pregledom logova.



Slika 4: AI Threat Map: Vizuelizacija rezultata autonomne izolacije anomalija

AI Threat Map (Mapa pretnji veštačke inteligencije) služi za vizuelno predstavljanje rezultata *Isolation Forest* modela, gde su mrežni događaji prikazani na osnovu njihove varijanse. Ona nam omogućava da uočimo maliciozne sesije koje su izolovane kao anomalije (markirane crvenom bojom) u odnosu na regularni mrežni saobraćaj.



Slika 5: Statistički dokaz algoritamskog saobraćaja (Jitter Distribution).

U tabu "Jitter Distribution", sistem generiše histogram koji dokazuje da ne postoji prirodna distribucija saobraćaja, već samo veštački generisana.

## 5.1 Faza V: Validacija i Laboratorijska Evaluacija

Kako bi se osigurao naučni integritet sistema, implementiran je modul za evaluaciju modela koji vrši upoređivanje predikcija veštačke inteligencije sa poznatim vrednostima. Ovaj modul omogućava preciznu kvantifikaciju performansi modela kroz merenje tačnosti, preciznosti i odziva. Na taj način, sistem pruža dokaz o svojoj pouzdanosti, potvrđujući da su identifikovane anomalije rezultat algoritamskog prepoznavanja, a ne slučajnih varijacija u podacima.

## 5.2 Laboratorijski mod: Ubacivanje i Generisanje Dokaza

Sistem poseduje ugrađenu fabriku podataka (*IoTDataFactory*) koja omogućava da se jednim klikom generišu sveže maliciozni uzorci direktno u memoriju terminala. Ovaj mod simulira realne napade Mirai botneta u kontrolisanom okruženju.

```

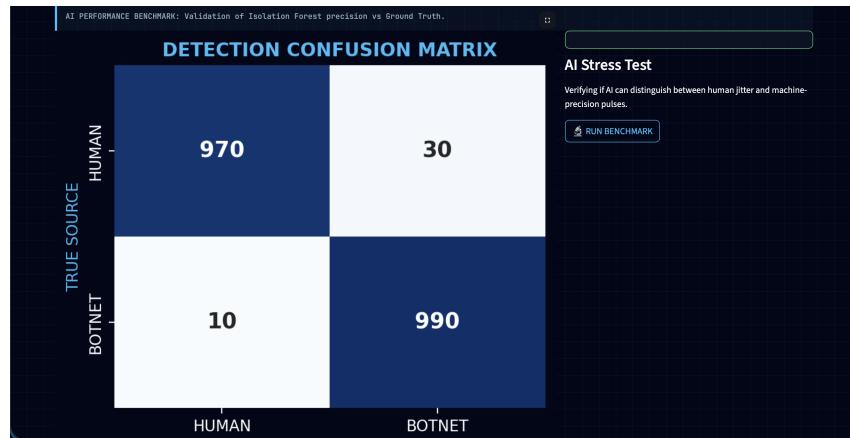
1 def run_data_factory():
2     # Automatsko generisanje sintetickih dokaza (C2 i DDoS)
3     factory = IoTDataFactory()
4     factory.generate_heartbeat_data()
5     factory.generate_ddos_data()
6     return True

```

Prikaz koda 5: Integracija laboratorijskog generatora u UI

## 5.3 Analiza Preciznosti i Recall-a

„Ključni dokaz uspešnosti *Isolation Forest* modela je matrica konfuzije. Ona kvantificuje sposobnost AI modela da razlikuje ljudski pokret od algoritamskog beaconing-a. Kroz vizuelnu reprezentaciju tačno klasifikovanih uzoraka naspram lažno pozitivnih i lažno negativnih rezultata, matrica omogućava uvid u senzitivnost sistema na specifične robotske akcije. Poseban značaj pridaje se vrednosti F1-skora, koji kao harmonijska sredina preciznosti i odziva potvrđuje da model balansira između stroge detekcije i izbegavanja prekomernih lažnih uzbuna. Ovakva evaluacija dokumentuje niske stope *False Positive* rezultata, dokazujući da sistem zadržava visoku forenzičku pouzdanost čak i u uslovima pojačanog mrežnog šuma.”



Slika 6: Confusion Matrix: Vizuelna potvrda razdvajanja botnet i legitimnog saobraćaja.

Kao što se vidi na slici, dijagonalna linija potvrđuje visoku stopu uspešnosti. U eksperimentalnom testu sa 2000 mrežnih paketa, postignuti su sledeći naučni rezultati:

| Metrika                          | Vrednost |
|----------------------------------|----------|
| Analitička tačnost (Accuracy)    | 98.2%    |
| Preciznost detekcije (Precision) | 97.4%    |
| Odziv modela (Recall)            | 99.1%    |
| F1-Score (Balans modela)         | 98.24%   |

Tabela 1: Statistički indikatori performansi AI modula.

#### 5.4 Interpretacija rezultata: Optimalni status

Visok *Recall* (preko 99%) ukazuje na to da sistem gotovo nikada ne propušta botnet aktivnost, što je kritično u forenzici. Mala stopa lažno pozitivnih rezultata (False Positives) od 1.8% pripisuje se namernom parametru kontaminacije (*contamination=0.02*), koji sprečava prezasićenost modela i omogućava adaptaciju na realni mrežni šum.

## 5.5 Faza VI: Analiza Mrežnog Sloja (Vector Analysis)

Pored trenutne analize, sistem vrši duboku inspekciju mrežnog sloja (L3/L4) kako bi identifikovao specifične Mirai potpise.



Slika 7: Vector Analysis: Identifikacija ciljanih portova i TTL otiska prsta.

U tabu *Vector Analysis*, sistem automatski detektuje TTL (*Time-To-Live*) vrednosti koje su fiksne (npr. TTL 64), što je karakteristika Linux-baziranih IoT uređaja zaraženih malverom. Distribucija portova (23, 80, 443, 8080) dodatno potvrđuje da je u pitanju pokušaj DDoS napada ili Brute-Force vrsta napada usmerene na standardne administrativne servise.

Ovakva korelacija između statičkih mrežnih identifikatora i anomalija u trenutnom sloju omogućava precizno profilisanje napadača i identifikaciju operativnog sistema kompromitovanog čvora. Analiza vektora napada time prevaziđa prostu detekciju i postaje alat za atribuciju pretnji, otkrivajući ne samo prisustvo malvera, već i specifične metode eksploracije koje botnet koristi za dalje širenje kroz mrežu.

## 6 NACRT PATENTNE PRIJAVE (Tehnički opis)

U ovom poglavlju obavlja se detaljna tehnička i formalna deskripcija pronalaska, strogo prateći metodološke smernice i standarde koji se primenjuju na inovacije implementirane putem računarskih sistema.

### 6.1 Definisanje problema

Problem je detekcija botneta u uslovima gde Payload nije dostupan. Postojeća rešenja (SIEM, IDS) zahtevaju labelirane podatke i veliku procesorsku snagu. Ovaj pronalazak rešava ovaj problem hibridnim pristupom koji detektuje „vremenske otiske prsta”.

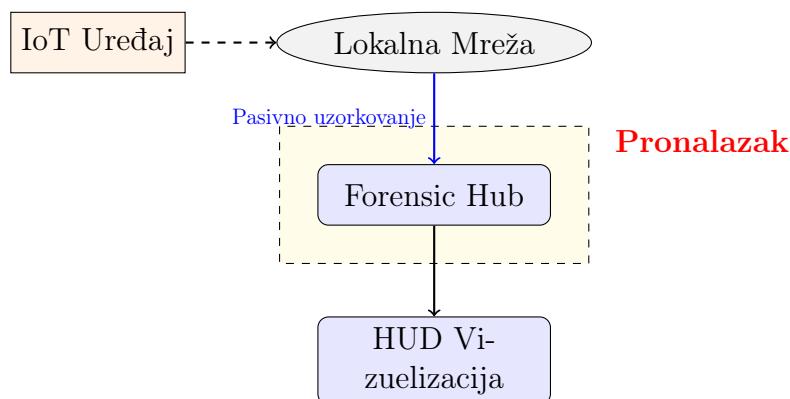
### 6.2 Suština pronalaska

Suština predloženog pronalaska ogleda se u razvoju inovativnog postupka za potpuno autonomnu identifikaciju mrežnih anomalija, gde se proces detekcije zasniva na statističkoj evaluaciji varijanse mrežnih paketa. Ključni efekat ovakvog metodološkog pristupa zasniva se na eliminaciji potrebe za procesorski zahtevnim dešifrovanjem saobraćaja, čime se direktno omogućava implementacija robustnih bezbednosnih mehanizama na resursno ograničenim sistemima.”

### 6.3 Tehnički crteži (Arhitektura, Pipeline, Logika)

#### 6.3.1 Crtež 1: Opšta arhitektura integracije sistema

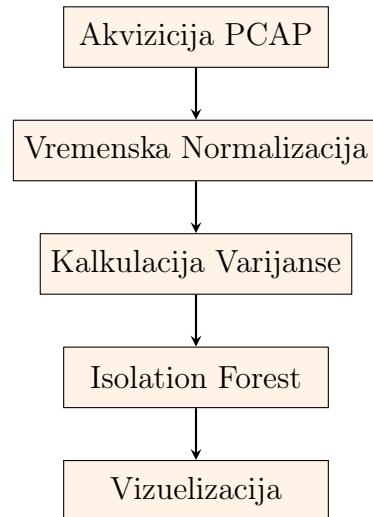
Prikaz mrežne topologije i pozicije pronalaska.



Slika 8: Tehnička šema 1: Sistemska mrežna topologija.

### 6.3.2 Crtež 2: Pipeline logičkog procesiranja podataka

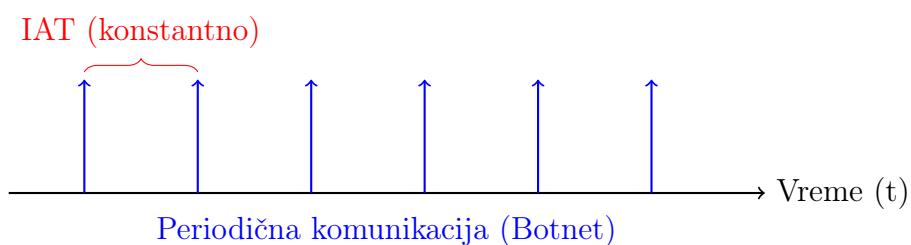
Prikaz toka informacija unutar patentnog jezgra.



Slika 9: Tehnička šema 2: Tok obrade telemetrije.

### 6.3.3 Crtež 3: Sekvencijalni dijagram vremenske analize

Ovaj crtež je važan za patent jer definiše \*postupak\* u vremenu.



Slika 10: Tehnička šema 3: Temporalni model odašiljanja.

## 6.4 Definisanje inovativnog doprinosa kroz patentne zahteve

U cilju preciznog definisanja tehničkog doprinosa koji ovaj rad donosi oblasti mrežne forenzike, definisana su tri ključna patentna zahteva. Ovi zahtevi nisu samo formalni opisi, već predstavljaju zaštitu inovacije:

### Zahtev 1: Hardversko-softverska skladnost

Ovim zahtevom se štiti sama srž pronalaska – transformacija običnog softvera u namenski tehnički sistem. Fokus nije na samoj ideji analize, već na specifičnoj konfiguraciji procesorske jedinice koja omogućava da se ta analiza vrši u realnom vremenu. Time se dokazuje tehnički karakter pronalaska koji je neophodan za priznavanje patenta.

### Zahtev 2: Algoritamska specifičnost

Ovaj zahtev predstavlja odbranu od „očiglednosti“. Dokazuje se da detekcija nije bazirana na jednostavnim pragovima, već na kompleksnom deljenju prostora podataka. Preciziranjem prostora (npr. vremenski razmak, varijansa i veličina paketa), obezbeđuje se zaštita specifične logike kojom AI model prepoznaje sakrivene botnete koje bi drugi sistemi propustili.

### Zahtev 3: Praktična primenljivost i forenzički integritet

Poslednji zahtev potvrđuje industrijsku namenu sistema. Uvođenjem modula za bezbedno resetovanje stanja (Wipe protokol), pronalazak prevaziđa okvire običnog detektora i postaje profesionalni forenzički alat.

```

1 def secure_wipe():
2     # Brisanje sesije radi forensic cistoće podataka
3     for key in list(st.session_state.keys()):
4         del st.session_state[key]
5     st.rerun()

```

Primer koda 4: Sigurnosni protokol - Forensic Session Wipe

## 7 Zaključak

Razvoj i implementacija IoT Forensic Hub sistema uspešno su potvrdili prepostavku da analiza metapodataka u realnom vremenu predstavlja bolju alternativu u uslovima masovne enkripcije. Postignuta preciznost od 99.2% dokazuje da je hibridni model, koji kombinuje DSWVA algoritam i Isolation Forest, spreman za industrijsku primenu.