

Che tipologie di test posso fare con Rita per un covert channel icmp. È per la tesi

Thought for 29s >

Ottima domanda — per una tesi su un covert-channel ICMP con RITA puoi progettare una batteria di test che verifica quando RITA rileva (**beaconing / long connections / strobes**) e **in quali condizioni invece perde il canale**. Qui sotto ti do (1) un elenco di tipi di test concreti, (2) una procedura sperimentale passo-passo con comandi RITA utili, e (3) metriche da raccogliere e interpretare. Ho incluso riferimenti alle pagine ufficiali / guide su RITA e a lavori su rilevamento ICMP per supportare le raccomandazioni. [activecm.github.io](http://activecm.github.io) +4

## 1) Tipologie di test (con obiettivo)

- **Beaconing (timing) tests** — invia pacchetti ICMP a intervalli regolari (es. every 5s) e poi con jitter ( $\pm 10\text{--}50\%$ ) e randomizzazione; verifica se RITA segnala beacon score elevati. (Obiettivo: quando l'algoritmo di RITA sulla periodicità coglie il pattern).

Black Hills Infor...

- **Throughput / payload-size tests** — prova trasferimenti con payload contenuti in diversi chunk (es. 64B, 256B, 512B, 1500B) e file totali di varie dimensioni (10 KB, 100 KB, 1 MB, 3 MB). (Obiettivo: capire se grandi o molteplici pacchetti appaiono come "long connections" o triggerano IDS/filtri). Trisul Network A...
- **One-way vs Two-way** — testa solo ICMP Echo Request (unidirezionale) vs request+reply con ACK/logica applicativa (bidirezionale). RITA si basa su connessioni osservate: cambia il profilo. [activecm.github.io](http://activecm.github.io)
- **Timing-channel (covert timing) vs payload-channel** — fingerprinta la detection se nascondi dati nella tempistica (es. moduli di inter-packet) invece che nel payload. RITA è forte su periodicità/frequenza ma meno su steganografia temporale molto sottile.

Active Counter...

- **Entropy / criptazione payload** — confronta payload ASCII prevedibile vs payload cifrato/alto entropia: alcuni rilevatori cercano pattern ripetuti o dimensioni coerenti.

Trisul Network A...

- **Mimicry tests** — genera ping che imitano traffico legittimo (e.g., uso di packet sizes e intervalli usati dai client normali o dallo streaming di monitor), per valutare quanto è facile evadere RITA. Active Counter...
- **Fragmentazione / IP options / TTL changes** — frammentare il payload o cambiare TTL flags per vedere se i log/Zeek pcap normalizzano o se RITA perde visibilità.
- **Background noise / scale tests** — esegui i test su pcap "puliti" e su pcap con traffico reale di rete (baseline) per valutare false negatives/positives. RITA lavora meglio con baseline e whitelisting. [activecm.github.io](http://activecm.github.io)

## 2) Procedura sperimentale raccomandata (passo-passo)

1. **Ambiente controllato:** usa due host (sender/receiver) dentro una VM o rete di laboratorio. Cattura tutto in pcap (tcpdump/wireshark) e/o metti Zeek a fare logging live. (Zeek → RITA è il flusso raccomandato). Active Counter...
2. **Genera i pcaps / log:** per ogni scenario (vedi matrice sotto) genera 1–2 minuti o più di traffico, assicurandoti di includere traffico di "rumore" per alcuni scenari.

**3. Produc Zeek logs** (conn.log, icmp.log, dns.log se usi DNS ecc.). Se parti da pcap:

zeek -r trace.pcap per ottenere i log da dare a RITA. Black Hills Infor...

**4. Importa in RITA:**

- rita import /path/to/zeek/logs dataset\_name

- rita analyze dataset\_name (fa le analisi/metriche interne). Black Hills Infor...

**5. Comandi RITA utili (output da salvare come CSV):**

- rita show-beacons dataset\_name -H → lista host con punteggio beacon.

- rita show-long-connections dataset\_name -H → connessioni lunghe (utile se usi trasferimento dati).

- rita show-strobes dataset\_name -H → strobe/fast beacon detection.

- rita html-report dataset\_name → report completo per ispezione visuale.

(Puoi esportare CSV per analisi numerica e calcolare TP/FN). activecm.github.io +1

**6. Valuta:** per ogni scenario calcola — rilevato da RITA? (Si/No), beacon score, lunghezza connessione, intervallo medio e deviazione standard, entropia del payload (puoi calcolarla off-RITA).

**7. Ripeti** cambiando un solo parametro per test (principio dell'esperimento controllato).

### 3) Matrice di test (esempio compatto)

Test ID	Payload size	Intervallo	Jitter	Fragmentation	Background
A1	64B chunks, file 10KB	5s fixed	0%	no	none
A2	64B	5s	±20%	no	none
B1	512B, file 1MB	5s	0%	no	none
B2	512B	5s	±20%	yes	none
C1	1500B	random intervals (poisson)	high	no	real traffic
D1	128B encrypted	5s	0%	no	none
E1	timing channel (encode bit in delay: 1s/2s)	variable	—	no	none

(Adatta il numero di ripetizioni, durata e pcap raccolti.)

### 4) Cosa guardare nei risultati e metriche

- Beacon score / ranking** (output show-beacons ) — se il tuo IP appare in top N.

activecm.github.io

- False negatives** — scenari in cui sai che c'è covert traffic ma RITA non lo segnala. Documenta parametri esatti.

- False positives** — traffico benigno che RITA classifica come sospetto (valuta rumore di rete).

- **Sensibilità rispetto a:** intervallo, jitter, payload size, entropia, frammentazione. Cerca soglie (es. sotto X ms jitter RITA rileva, sopra Y non rileva).
- **Visual analysis:** usa `rita html-report` per grafici e timeline e confronta con i raw pcap/Zeek logs. [activecm.github.io](https://activecm.github.io)

## 5) Ipotesi e punti di attenzione per la tesi

- RITA è progettato per trovare **beaconing** e **connessioni anomale** (frequency/time analysis): se il tuo covert channel è fatto per **mimicry** (es. copia esatta del pattern di ping legittimi) allora può eludere RITA — documenta quest'aspetto. [Active Counter...](#)
- Canali di **solo payload** che dividono un file in tanti ping piccoli potrebbero non essere visti come "beacon" ma come "long connections" o potrebbero passare inosservati se ben randomizzati; quindi testa entrambe le modalità. [Trisul Network A...](#)
- Valuta anche altri strumenti complementari (AC-Hunter, Suricata/IDS rules per ICMP size anomalies) per avere un confronto multilivello. [Active Counter...](#)

## 6) Esempi concreti di comandi / output da includere nella tesi

- Import + analisi:

pgsql

 Copia codice

```
rita import /home/user/zeek-logs mytestA1
rita analyze mytestA1
rita show-beacons mytestA1 -H > mytestA1_beacons.csv
rita show-long-connections mytestA1 -H > mytestA1_longconns.csv
rita html-report mytestA1
```

- Salva i CSV e crea grafici (intervallo medio, deviazione, beacon score rank) per ogni scenario.

## 7) Idee aggiuntive di analisi (valorizzano la tesi)

- **ROC curve:** considera RITA come classificatore (soglia  $\delta$ ) e calcola true/false positive al variare della soglia → curva ROC/PR.
- **Feature-importance:** estrai per ogni connessione gli indicatori che RITA usa (intervalli, variazione, lunghezza) e dimostra quali parametri più influenzano la rilevazione.
- **Confronto strumenti:** esegui gli stessi pcaps su RITA, Suricata/IDS e un semplice script di statistica per vedere differenze di detection.

Se vuoi, preparo subito per te (posso farlo ora):

- una **matrice di test** completa in CSV pronta da eseguire (ID scenario, parametri, durata), e/o
- degli **script di generazione pcap** (es. Python/scapy) per i principali scenari (A1, A2, B1, C1) in modo che tu possa rigenerare i test ripetutamente, e/o
- un template di **sezione metodi** per la tesi che descriva setup, comandi RITA e come calcoli TP/FN (pronto da incollare).

Dimmi quale vuoi che generi e te lo preparo subito (script + matrice + comandi). Non chiedermi di aspettare: lo produco ora.

     ...     [Fonti](#)