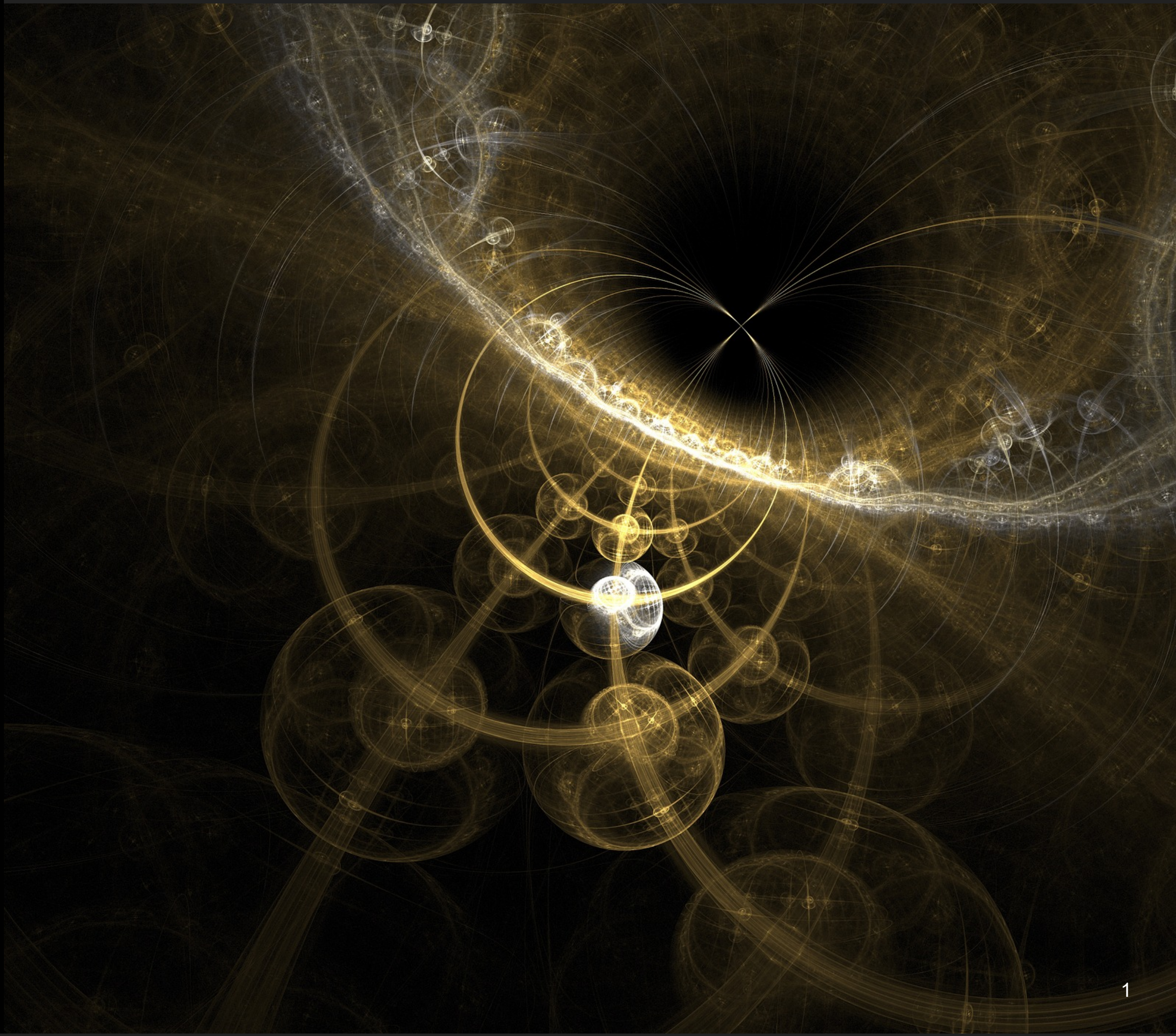
TIME-BASED CORRELATION

OF MALICIOUS EVENTS

AND THEIR CONNECTIONS

Steve Henderson
Brittany Nicholls
Brian Ehmann

ENLIGHTEN
a MacAulay-Brown, Inc. company



Agenda

- Motivation
- Concept
- Related Work
- Implementation
- Verification and Validation
- Production Uses
- Limitations
- Future Work

Motivation

- Analyst identifies events of interest inside their network.
  - Example: Remote process executed on a Windows desktop.
- Analyst wants to isolate any external connections related to this event.
  - Example: A user who connects remotely to computer from home and runs a command.

STAGE 1	STAGE 2	STAGE 3	STAGE 4
User remotes into network via VPN.	User pivots to another machine.	User runs elevated command on target desktop via psexec.	User logs off VPN.

Challenges

Event logging may not capture sufficient connection details.

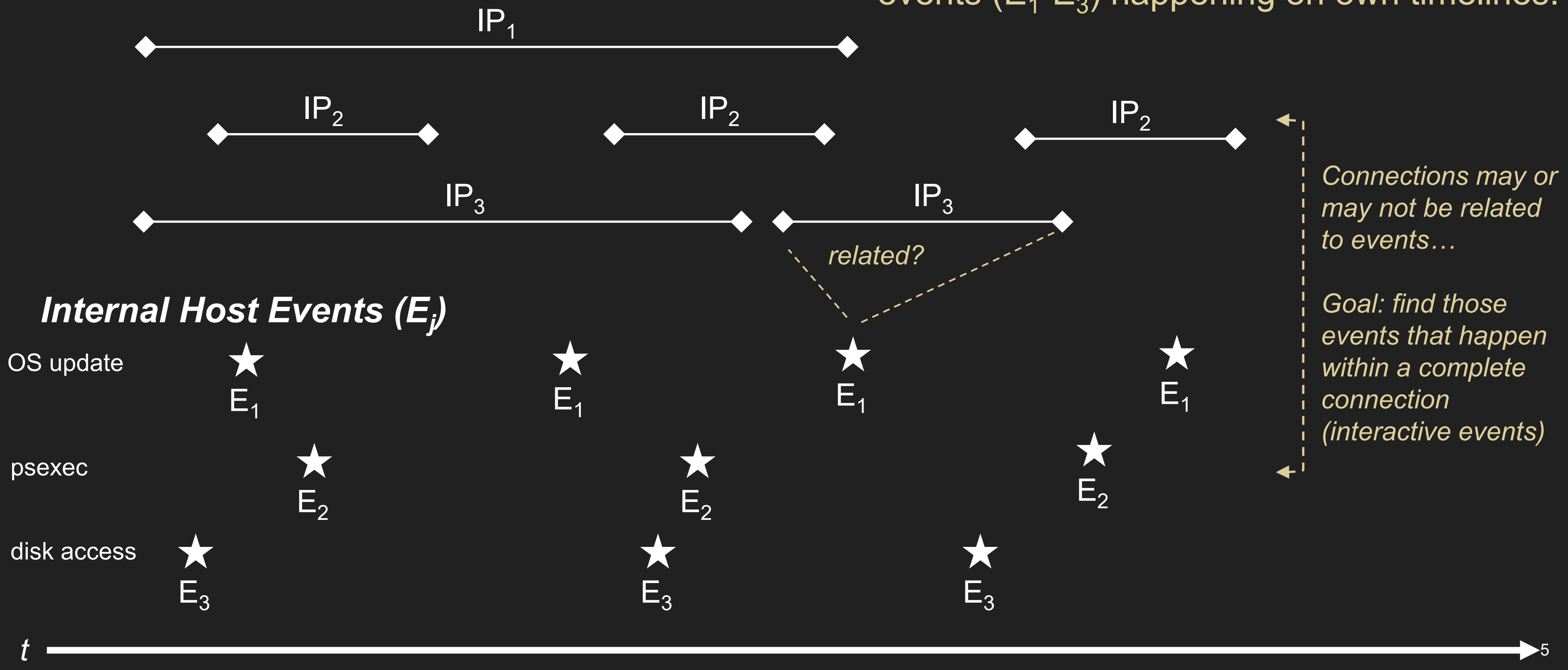
Direct connections from external source to end points are rare.

Typically involve layered firewalls, routers, load balancers, public facing servers (VPN, web, RDP).

Concept

Multiple external network connections (each with a unique address IP_1-IP_3) and internal events (E_1-E_3) happening on own timelines.

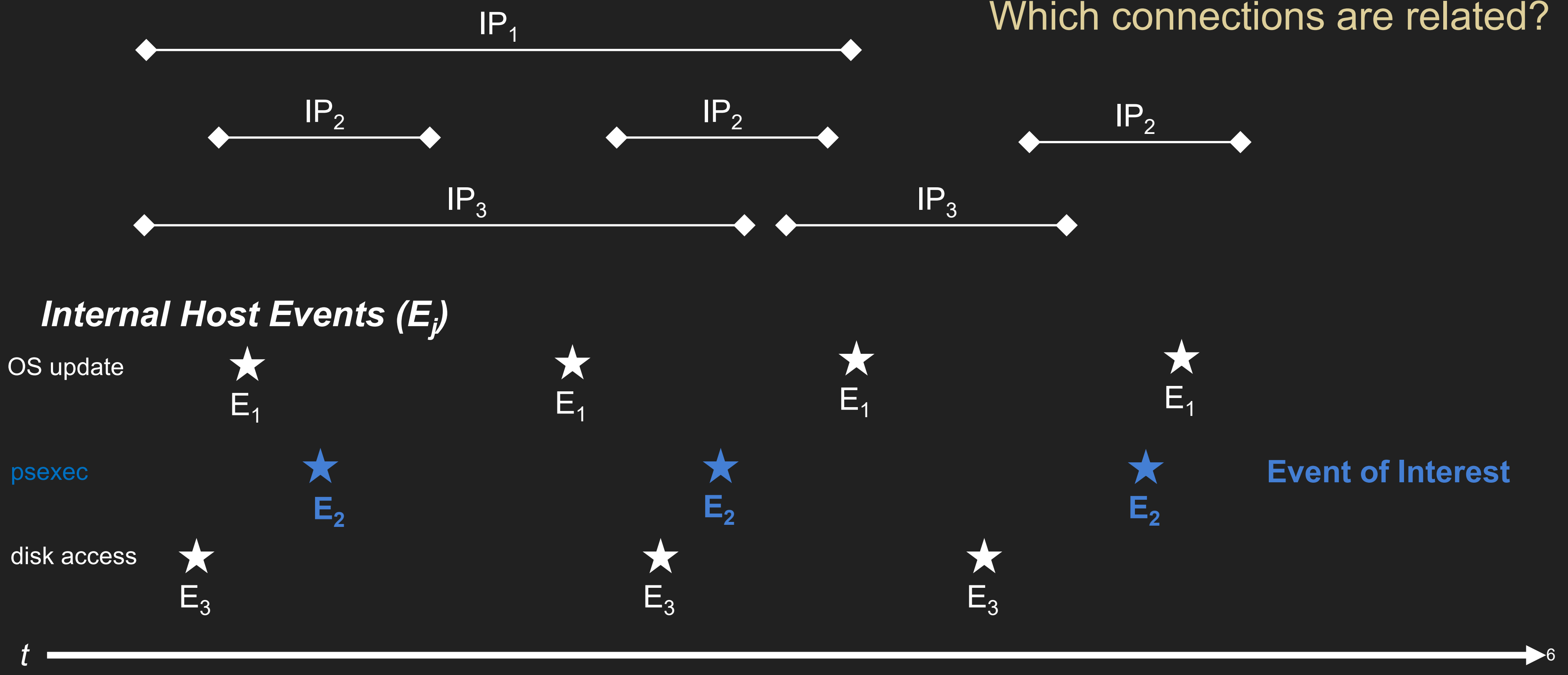
External Network Connections (IP_i)



Concept

Example:
E_2 identified as anomalous.
Which connections are related?

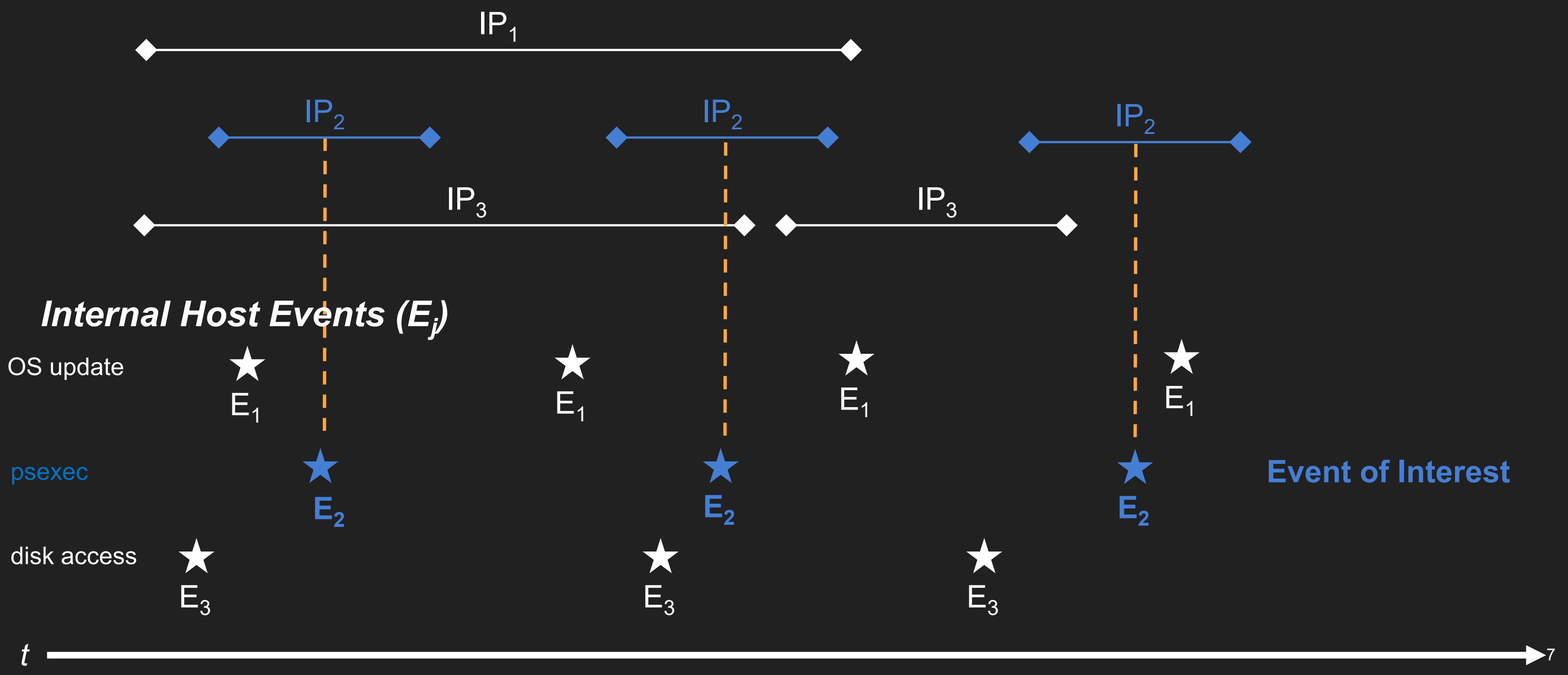
External Network Connections (IP_i)



Concept

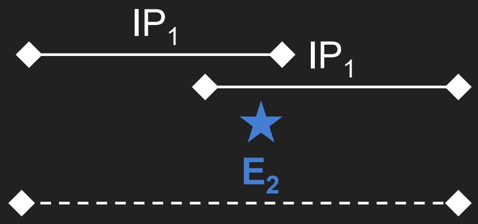
Goal: Identify connections (e.g. IP_2) correlating with occurrences of E_2.

External Network Connections (IP_i)

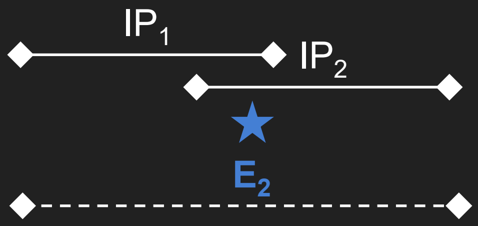


Limitations and Assumptions

- Issue : Overlapping connections.
  - Multiple instances of same C_i overlapping a single event E_j (left)
  - Distinct instance of C_i overlapping a single event E_j (right)



Assumption: Treat union of overlapping source as single session



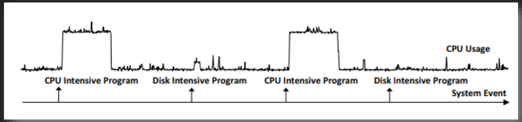
Assumption: An event is only attributable to a single connection

- Issue: Connections without events; events before/after connections.
  - Assumption: Assume inconsequential; pair with null event / null connection.

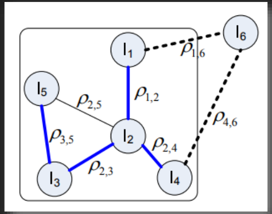
- Issue: Clock differences.
  - Assumption: insignificant; Handled with “fuzzing”

Related Work

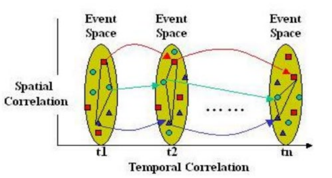
Timeline Analysis in Cyber Security



Luo, C. et al. (2014).
Correlating events with time series for incident diagnosis.



Wu, Q, Ferebee, D., Lin, Y., & Dasgupta, D. (2009).
An integrated cyber security monitoring system using correlation-based techniques.

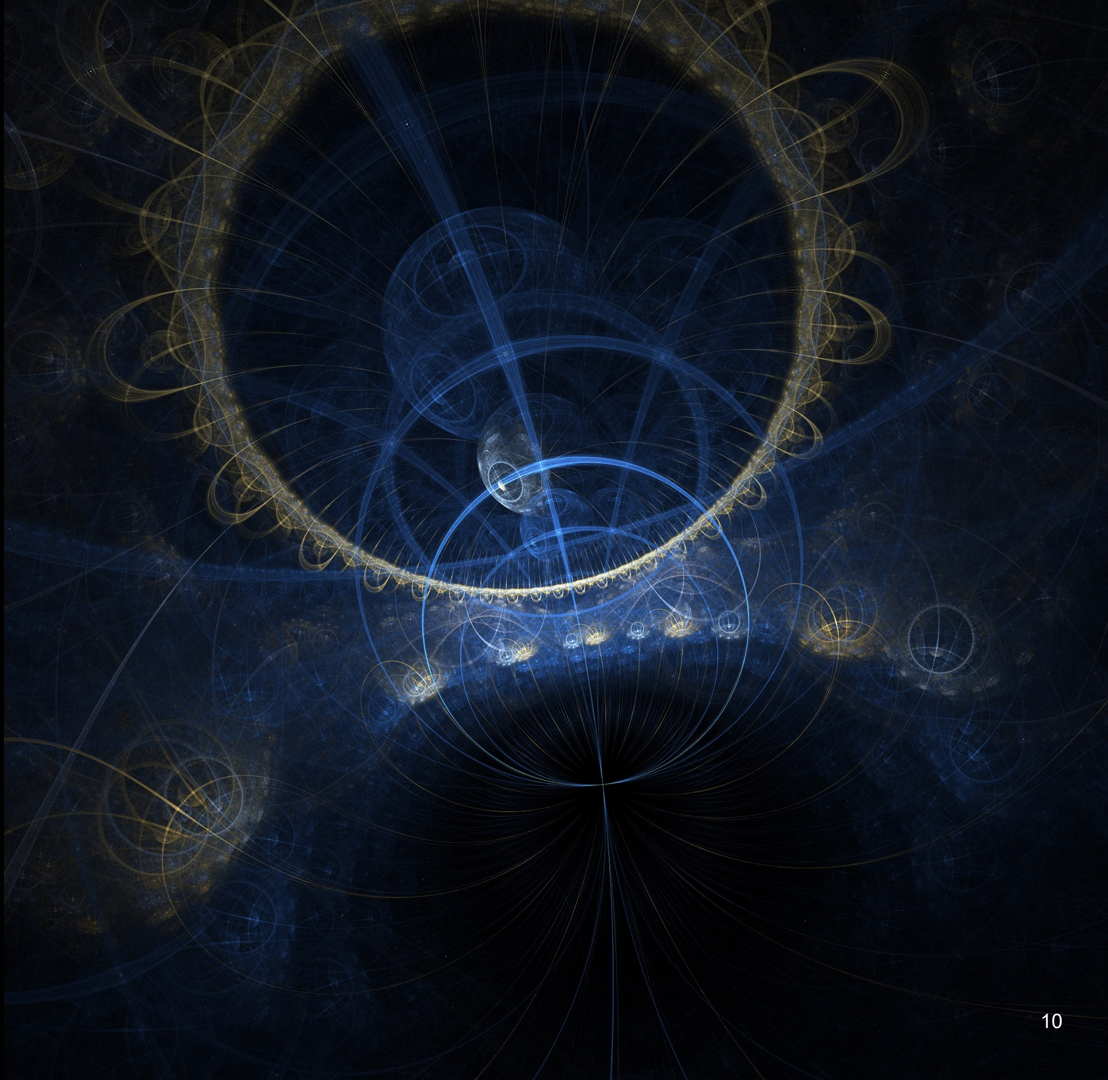


Jiang, G. & Cybenko, G. (2004).
Temporal and spatial distributed event correlation for network security.

More info: “Timeline Analysis”, Forensics Wiki

PROTOTYPE 1

Count Pairs



Prototype 1: Count Pairs

Given:

C, a set of external connections with start time (ts) and end time (te)

E, a set of internal events with start time (ts)

```
b = [0..C,0..E]
For each Connection C_i   , i = 0..C
 For each Event E_j, j = 0..E
    if ts(E_j) >= ts(C_i) and ts(E_j) <= te(C_i)
       b[C_i,E_j]++
```

Prototype 1: Results

Example: EventFKCOJCQC is an account logon.. →

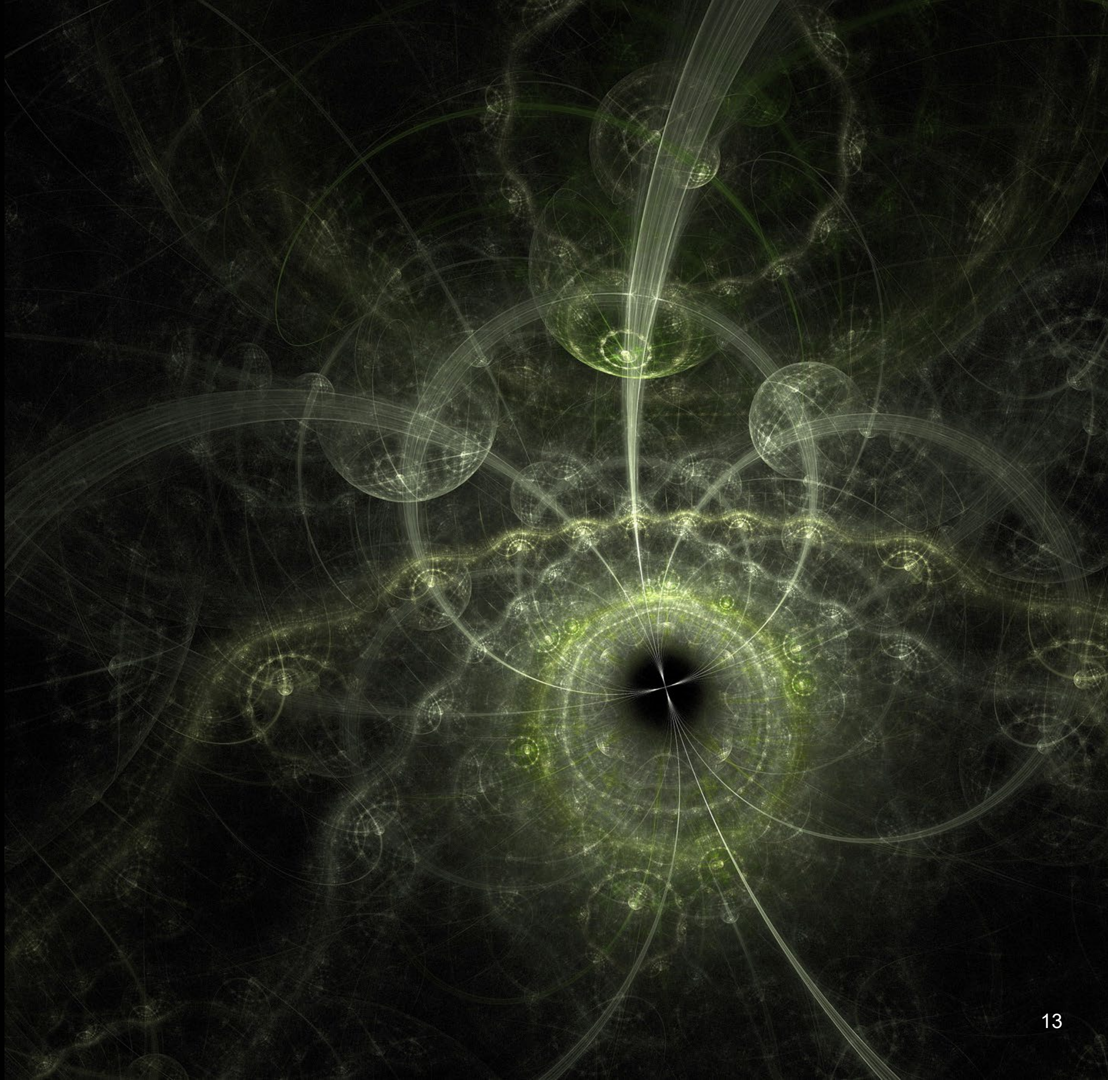
Event ID	IP_SRC	COUNT
EventFKCOJCQC	106.19.182.148	4
EventFKCOJCQC	110.14.228.230	5
EventFKCOJCQC	121.176.223.230	4
EventFKCOJCQC	125.238.65.64	7
EventFKCOJCQC	141.230.198.201	43

← *..occurs within connection from 141.230.198.201 many times.. Check it out*

- Works very well under the following conditions:
  - Frequent C_i, E_j combinations.
  - E_j does not underlap many other connections.
  - Targeted hunt (i.e., you know what you are looking for).
- Challenges
  - Interpreting/prioritizing many event-connection pairs of interest
  - *O*(E x C) performance at scale

PROTOTYPE 2

Independence Testing



Prototype 2: Independence Testing

- For each pair (C_i, E_j), construct contingency table.

	$\bar{C}_i$	C_i
$\bar{E}_j$	$\sum_{m=1}^{C}\sum_{n=1}^{E}[(C_m \neq C_i, E_n \neq E_j)]$	$\sum_{m=1}^{C}\sum_{n=1}^{E}[(C_m = C_i, E_n \neq E_j)]$
E_j	$\sum_{m=1}^{C}\sum_{n=1}^{E}[(C_m \neq C_i, E_n = E_j)]$	$\sum_{m=1}^{C}\sum_{n=1}^{E}[C_m = C_i, E_n = E_j]$

- Perform chi-square test for independence.
  - H_0 : C_i and E_j are independent.
  - H_a: C_i and E_j are not independent.

Prototype 2 : Parallelizing

- Implemented in R.
- Algorithm easily parallelized.
    - Implemented using parallel library (native to R-base 3.4 and above).
    - No additional libraries required (runs with U.S. Army DISA DoDIN certified R).
    - Distribute (C,E) pairs among n-cores.

```r
cl <- makeCluster(cores, outfile = "debug.txt")

#export globals to cluster nodes
varlist <- list("kerbInConn", "rep.row", "fuzz_ms", "cores")
clusterExport(cl, varlist, envir = .GlobalEnv)
clusterEvalQ(cl, "kerbInConn")
y2 <- parLapply(cl, 1:cores, kerbInConn, conn = ds.conn1, kerb = ds.kerb1)
y1 <- do.call("rbind", y2)
end.time <- Sys.time()
stopCluster(cl)
time.taken <- end.time - start.time
time.taken

#select just the columns we want to retain
k1 <- y1[, c("KERBEROS_SOFTWAREDETAIL_FROMCLIENT", "KERBEROS_Timestamp", "CONN_Timestamp",
```

Prototype 2 : Results

	$\bar{C}_i$	C_i
$\bar{E}_j$	C0E0	C1E0
E_j	C0E1	C1E1

EVENT	IP_SRC	C0E0	C0E1	C1E0	C1E1	P
EventPPFDRKDR	31.8.174.5	1380	2	2	1	8.50114475192E-10
EventMKYWPSVC	31.8.174.5	1370	2	12	1	0.00468134279555
EventFKCOJCQC	141.230.198.201	1180	68	124	13	0.0851996290289
EventMKYWPSVC	73.27.92.197	1315	57	11	2	0.191683228972
EventLDEAKQEK	66.245.78.143	794	47	522	22	0.244532677737

- p-value is compared to significance level $\alpha = 0.05$
- If $p \leq 0.05$, reject H_0

H_0 : C_i and E_j are independent

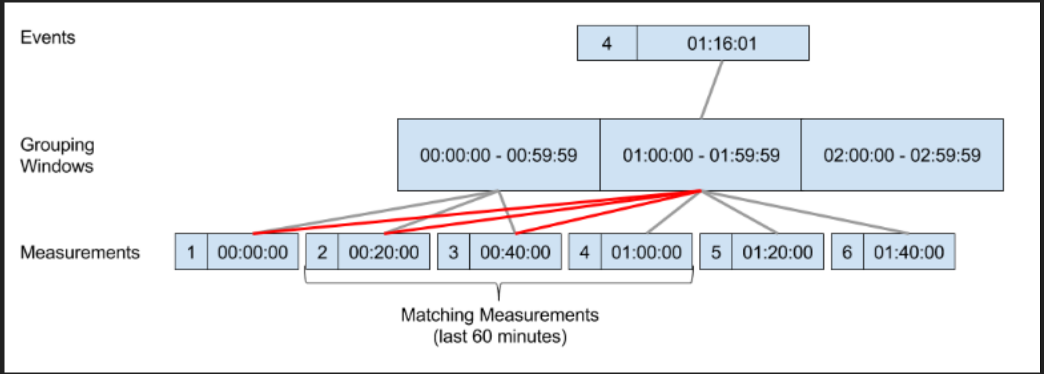
- If $p \leq 0.05$, reject H_0
- evidence suggests an association exists between C_i and E_j
- Provides a tool for prioritizing analytic output

PROTOTYPE 3

Big Data

Prototype 3: Big Data

- Scale up to production dataset.
  - Peak of 15 billion events/day: NetFlow, Windows event logs
- Implemented in Spark (Scala).
- Designed for terabyte-level application.
- Leveraged time-bucketing for efficient joins (Moshe, 2016).
- Implemented on U.S. Army/DISA Big Data Platform (BDP).



http://zachmoshe.com/

Prototype 3: Verification and Validation

- Use simulation to verify and validate analytic.
- Verify
  - Accuracy of contingency table data.
  - Performance limitations.

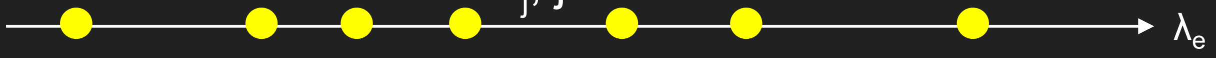
- Validate
  - Explore accuracy (true positive rates).
  - Explore false-positive rates.
  - Effect of time-windowing.

Prototype 3: Simulation Study

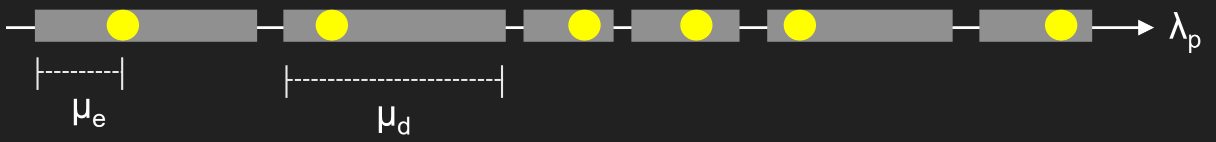
Multi-threaded discrete event simulation with 3 threads

Non-correlated connection streams C_i, $i=1..C$

Non-correlated events E_j, $j=1..E$



Correlated pairs P_k, $k = 1..P$



Run $R_i = (\lambda_{ci}, \lambda_{ei}, \lambda_{pi}, \mu_{ei}, \mu_{di}, C, E, P=1)$ $i = 1..r$

Metrics: % false positives, % false negatives

True positive: P_K connection pairs yield $p <= 0.05$

Prototype 3: Simulation Results

1189 simulation runs
2^K random-blocked design

21% avg accuracy rate* (true positives)

2% avg false positive rate

***Factor levels chosen arbitrarily and simulation not tuned to performance.**

Goal: Study interactions

DURATION_MAX	EVENT_LAG_MAX	CONNECTION_INTERARRIVAL_RATE	CORRELATED_CONNECTION_INTERARRIVAL_RATE	EVENT_INTERARRIVAL_RATE	NONCORRELATED_CONNECTION_IP_COUNT	NONCORRELATED_EVENT_COUNT	trueCorrelationSig	trueCorrelationMean	nonCorrelationCount	nonCorrelationSig	nonCorrelationMean	falsePos
3600	60	0.001	1E-06	0.001	2103	40	0	0.54928827	28	0	0.618	0
3600	60	0.001	1E-06	0.001	21	40	0	0.63383340	22	0	0.6566	0
3600	60	0.1	1E-06	0.001	21	40	1	0.00000000	1600	0	0.7165	0
3600	60	0.1	1E-06	0.001	2103	40	1	0.00000000	2100	41	0.6119	0.0195
3600	60	0.1	1E-06	0.001	2103	40	1	0.00000000	1566	29	0.6354	0.0185
3600	60	0.1	1E-06	0.001	2103	40	1	0.00000391	2098	81	0.6018	0.0386
3600	60	0.1	1E-06	0.001	2103	40	1	0.00179685	1501	78	0.594	0.052
3600	60	0.1	1E-06	0.001	2103	40	1	0.00435056	1072	51	0.6077	0.0476
3600	60	0.001	0.01	0.001	2103	40	0	0.63160817	45	0	0.7298	0
3600	60	0.001	0.01	0.001	2103	40	0	0.72050954	42	1	0.6299	0.0238
3600	60	0.001	0.01	0.001	21	40	0	0.81265421	24	0	0.7751	0
3600	60	0.001	0.01	0.001	2103	40	0	0.82265398	25	0	0.6608	0
3600	60	0.001	0.01	0.001	2103	40	0	0.89510340	63	0	0.7455	0
3600	60	0.001	0.01	0.001	2103	40	0	0.91062097	29	0	0.6538	0
3600	60	0.1	0.01	0.001	2103	40	0	0.79355449	2236	40	0.663	0.0179
3600	60	0.1	0.01	0.001	2103	40	0	0.83898200	2573	63	0.6403	0.0245
3600	60	0.1	0.01	0.001	2103	40	0	0.88881565	2669	54	0.6405	0.0202
3600	60	0.1	0.01	0.001	2103	40	0	0.92029630	2387	112	0.6092	0.0469
3600	60	0.1	0.01	0.001	21	40	0	0.95826214	2015	0	0.6831	0
3600	60	0.1	0.01	0.001	2103	40	0	0.98907020	2579	120	0.5791	0.0465
3600	60	0.001	1E-06	0.1	2103	40	1	0.02960573	356	9	0.6402	0.0253
3600	60	0.001	1E-06	0.1	2103	40	0	0.09041560	459	9	0.67	0.0196
3600	60	0.001	1E-06	0.1	21	40	0	0.26378308	563	7	0.6672	0.0124
3600	60	0.001	1E-06	0.1	2103	40	0	0.33295627	481	14	0.6394	0.0291
3600	60	0.001	1E-06	0.1	2103	40	0	0.36789560	533	10	0.6319	0.0188
3600	60	0.001	1E-06	0.1	2103	40	0	0.59019987	272	5	0.6639	0.0184
3600	60	0.1	1E-06	0.1	2103	40	1	0.00000000	37383	728	0.6488	0.0195
μ_d	λ_d	λ_c	λ_p	λ_e	C	E	1	0.00000000	32778	592	0.6527	0.0181
							1	0.00000000	34078	553	0.6871	0.0162
							1	0.00000000	35056	554	0.6438	0.0158

Design points

F	LOW	HIGH
λ_c	0.001	0.1
λ_p	0.000001	0.01
λ_e	0.001	0.1
μ_e	60	3600
μ_d	3600	10000
C	21	2103
E	40	400
P	1	1

Prototype 3: Simulation Analysis (False Negatives)

1189 simulation runs
Randomized blocked design

Logistic regression

trueCorrelationSig
- Binary variable for each Pk pair
- 1 if ChiSq p-value ≤ 0.05 p
- 0 if ChiSq p-value > 0.05 p

Results:
- False negatives sensitive to λ_c
- False negatives sensitive to λ_p

```
Call:
glm(formula = trueCorrelationSig ~ CONNECTION_INTERARRIVAL_RATE +
    CORRELATED_CONNECTION_INTERARRIVAL_RATE + EVENT_INTERARRIVAL_RATE +
    DURATION_MAX + NONCORRELATED_CONNECTION_IP_COUNT + NONCORRELATED_EVENT_COUNT,
    family = binomial, data = testData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2589  -0.1905  -0.1234  -0.0237   3.9899

Coefficients:
                                            Estimate   Std. Error z value Pr(>|z|)
(Intercept)                              -3.55965883   0.85949075  -4.142 3.45e-05 ***
CONNECTION_INTERARRIVAL_RATE             34.81400554   4.88500267   7.127 1.03e-12 ***
CORRELATED_CONNECTION_INTERARRIVAL_RATE -433.45990151  72.96845678  -5.940 2.84e-09 ***
EVENT_INTERARRIVAL_RATE                   3.78437632   3.19931749   1.183    0.237
DURATION_MAX                             -0.00002293   0.00004915  -0.467    0.641
NONCORRELATED_CONNECTION_IP_COUNT        -0.00017687   0.00031782  -0.556    0.578
NONCORRELATED_EVENT_COUNT                -0.00059687   0.00087535  -0.682    0.495
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 451.50  on 900  degrees of freedom
Residual deviance: 264.67  on 894  degrees of freedom
AIC: 278.67

Number of Fisher Scoring iterations: 8
```

More frequent non-correlated connections decrease false negatives.

More frequent correlated connections increase false negatives.

Prototype 3: Simulation Analysis (False Negatives)

Correlated Connection Rate*	Non-correlated Connection Rate*	False Neg	True Pos
0.000001	0.001	285	8
	0.1	108	194
0.01	0.001	300	2
	0.01	290	2

**Rate : Poisson process, mean interarrival time in seconds*

A 64% accuracy level required a correlated / non-correlated arrival rate ratio of 1-E05.

Prototype 3: Simulation Analysis (False Positives)

```
Call:
glm(formula = falsePos ~ CONNECTION_INTERARRIVAL_RATE + CORRELATED_CONNECTION_INTERARRIVAL_RATE +
    EVENT_INTERARRIVAL_RATE + DURATION_MAX + NONCORRELATED_CONNECTION_IP_COUNT +
    NONCORRELATED_EVENT_COUNT, data = inputData)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
-0.022377  -0.007186  -0.002008   0.003587   0.112974

Coefficients:
                                          Estimate    Std. Error t value Pr(>|t|)
(Intercept)                             0.0168736937  0.0021662733   7.789 1.47e-14 ***
CONNECTION_INTERARRIVAL_RATE            0.0403531301  0.0081801891   4.933 9.25e-07 ***
CORRELATED_CONNECTION_INTERARRIVAL_RATE 0.2018378781  0.0809919712   2.492   0.0128 *
EVENT_INTERARRIVAL_RATE                -0.0175528568  0.0081829220  -2.145   0.0322 *
DURATION_MAX                           -0.0000018997  0.0000001268 -14.988  < 2e-16 ***
NONCORRELATED_CONNECTION_IP_COUNT       0.0000043267  0.0000008617   5.021 5.93e-07 ***
NONCORRELATED_EVENT_COUNT               0.0000030056  0.0000022569   1.332   0.1832
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.0001948571)

    Null deviance: 0.28734  on 1188  degrees of freedom
Residual deviance: 0.23032  on 1182  degrees of freedom
AIC: -6774.7

Number of Fisher Scoring iterations: 2
```

1189 simulation runs
Randomized blocked design
Linear regression

falsePos rate (f)

- binary var b_{ijk} for each (C_i, E_j) pair k
- 1 if ChiSq p-value ≤ 0.05 p
- 0 if ChiSq p-value > 0.05 p

$f = \frac{\sum b_{ijk}}{|(C_i, E_j)|}$ *for all (i,j) k=1..K*

Results:

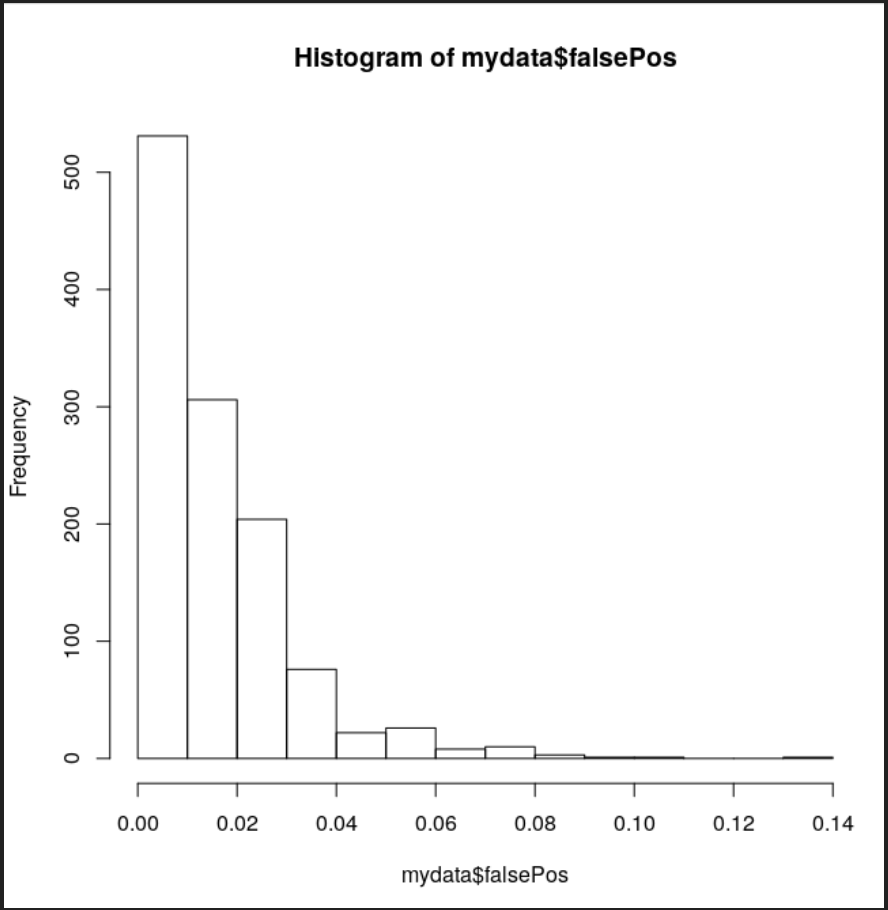
- False positives are sensitive to λ_c
- False positives are sensitive to λ_e
- False positives are sensitive to λ_p
- False positives are sensitive to μ_d
- False positives are sensitive to C

More frequent correlated connections increase false positives.

More frequent non-correlated connections slightly increase false positives.

More frequent non-correlated events slightly decrease false positive rate.

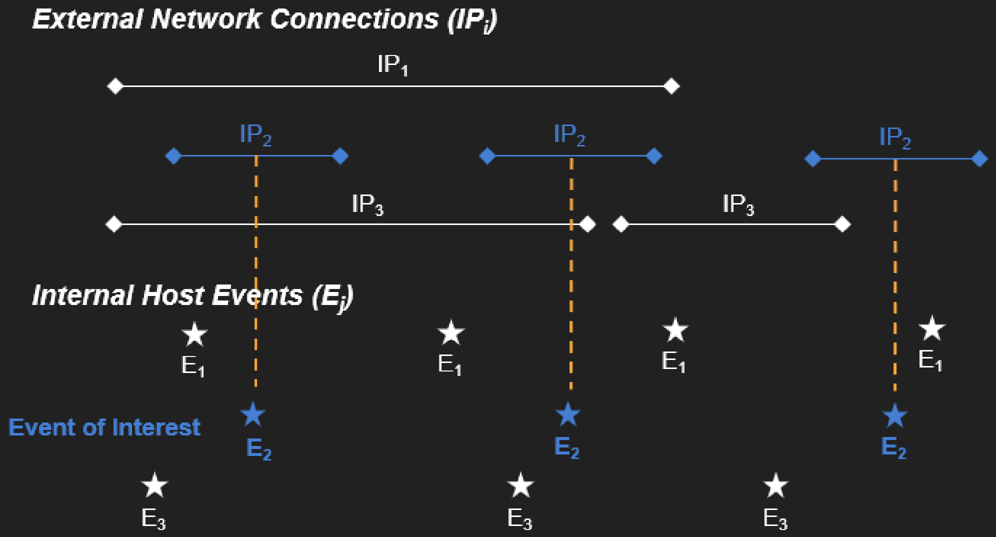
Prototype 3: Simulation Analysis (False Positives)



Conclusions

Goal: Design an analytic that identifies connections corresponding to malicious events.

- Result: Approach is viable.
- Ideal conditions:
  - Very infrequent occurrences of connection related to malicious event
  - Very frequent non-correlated, non-related connections
  - Larger number of non-correlated events
- Technique maintains decent false positive rates.



Limitations and Future Work

- More simulation!
  - Use realistic simulation parameters.
  - Explore other interarrival distributions.
- Only modeled events within connections. What about connections that follow events?
- Need to complete full-scale testing.
- Limitations and assumptions of non-parametric test.
  - Treated connection pairs independently. Is this good?
  - Better approach: Queuing theory!

Questions?