ClickMiner: Towards Reconstructing User-Browser Interactions from Network Traces

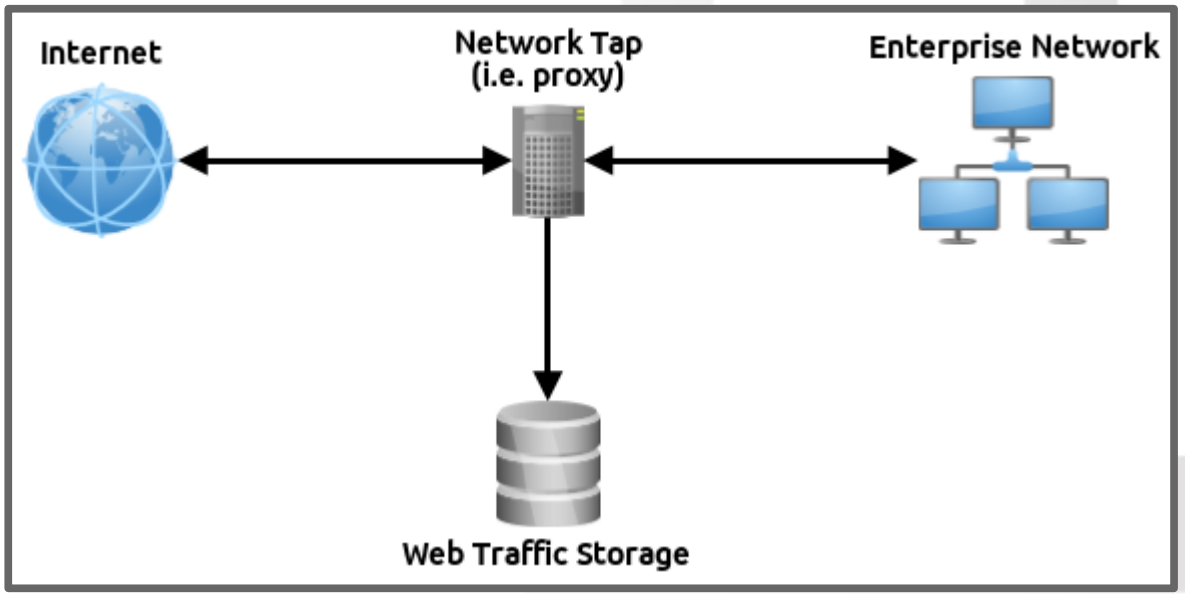
Christopher Neasbitt[†],
Roberto Perdisci[†‡], Kang Li[†], and Terry Nelms[⅂‡]
† Department of Computer Science, University of Georgia
‡ College of Computing, Georgia Institute of Technology
⅂ Damballa, Inc.

cjneasbi@uga.edu, {perdisci,kangli}@cs.uga.edu, tnelms@gatech.edu

Network Security Intelligence

Application Scenario

- An enterprise has made an investment to record a window of ingress and egress network traffic from it's local network
- This data could aid in the forensic investigation of security incidents
  - ex. Web Security Incidents
    - Phishing, Social Engineering, Data Leakage
- How do we effectively analyze such large amounts of web traffic?



Security Network Intelligence

Application Scenario

The modern web is becoming increasingly complex.

Increasing *Semantic Gap* between network traffic and user actions

Security Network Intelligence

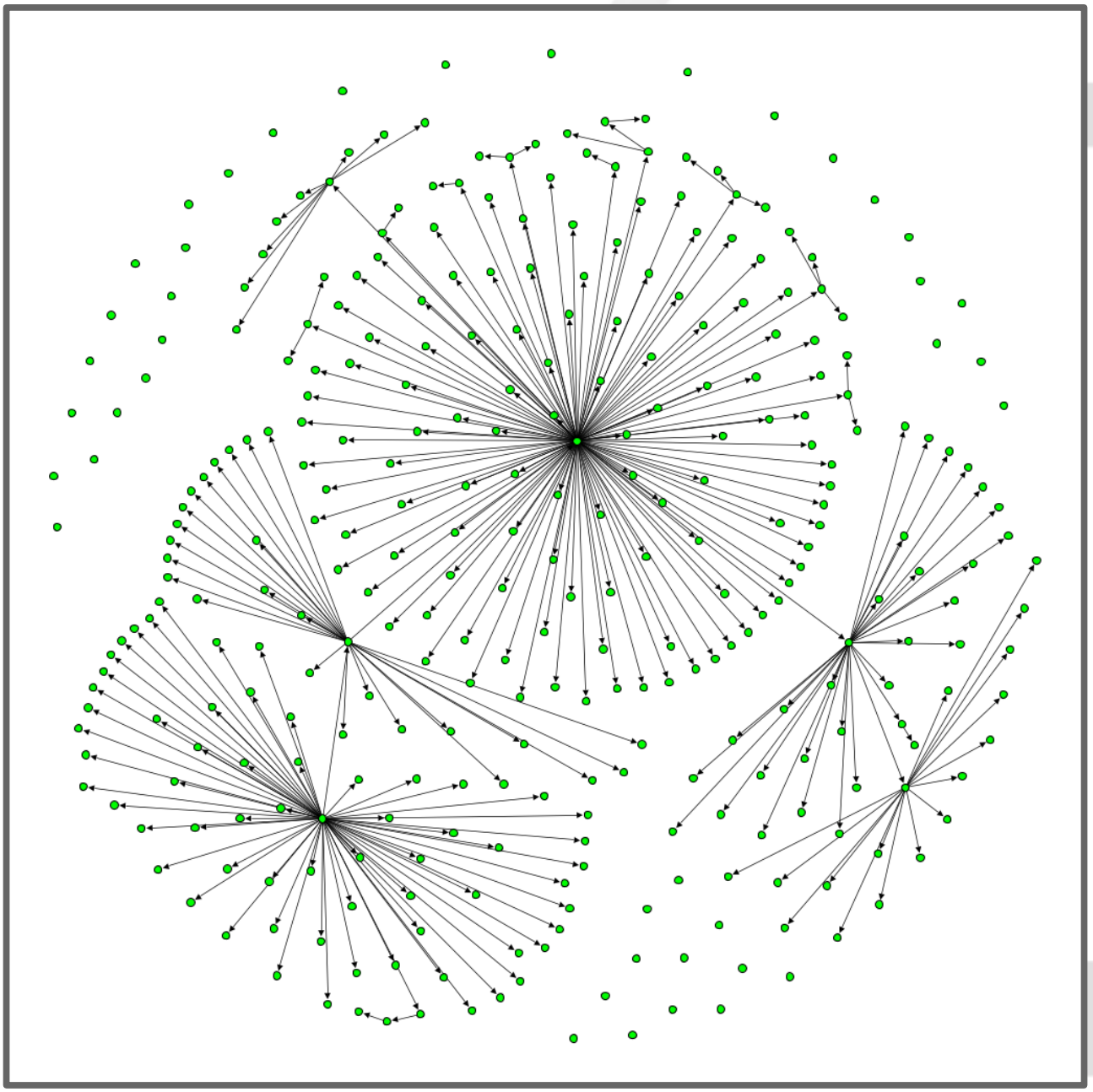
Application Scenario

Social Engineering Malware Download Scenario

- User searches for a crack to a particular game
- User is directed to a dubious site from the search results
- User downloads and installs a binary infected with malware from said site

Application Scenario

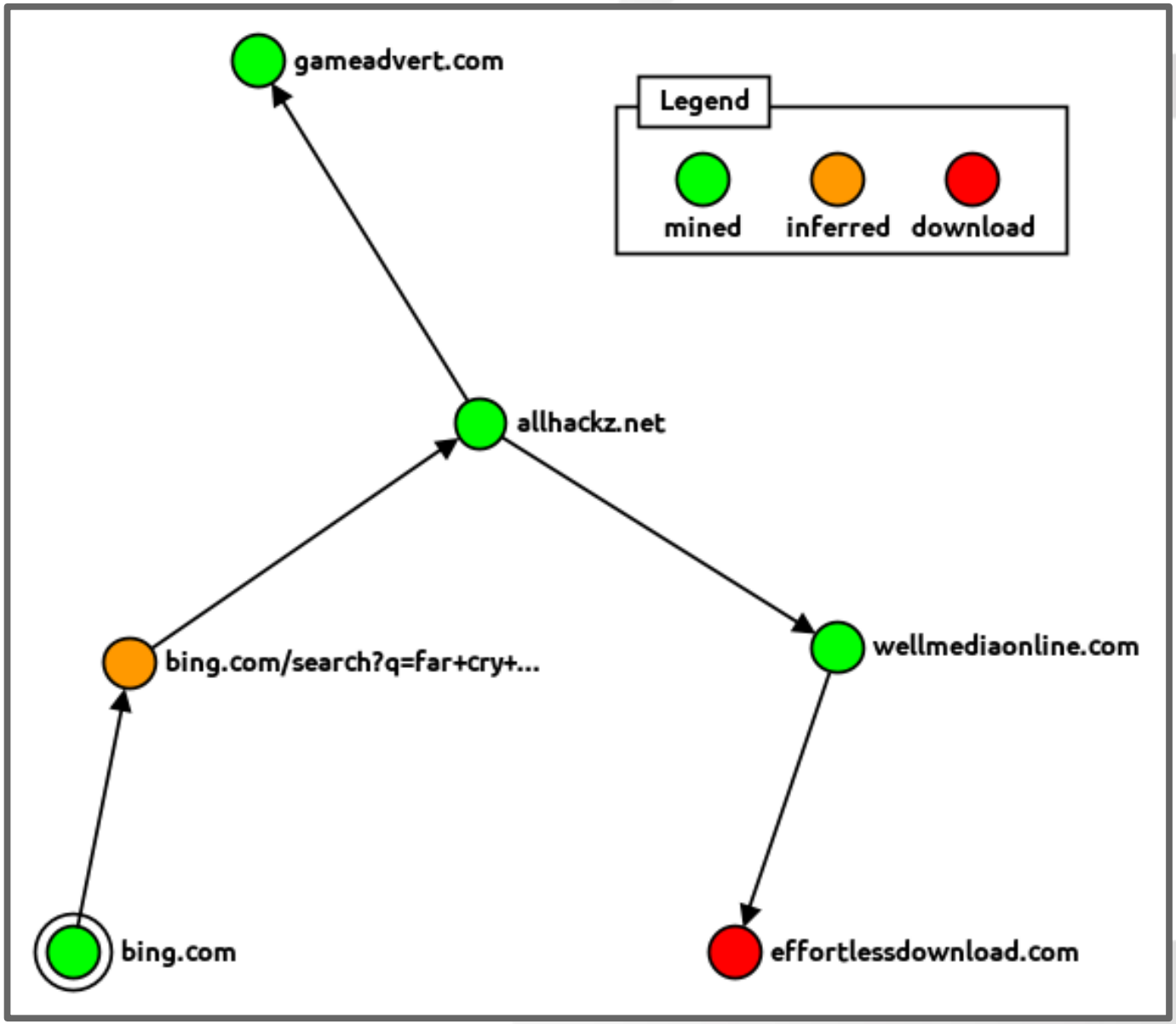
Requests: 328
Edges: 287
Clicks: 6
Time: 2m 56s



Security Network Intelligence

Application Scenario

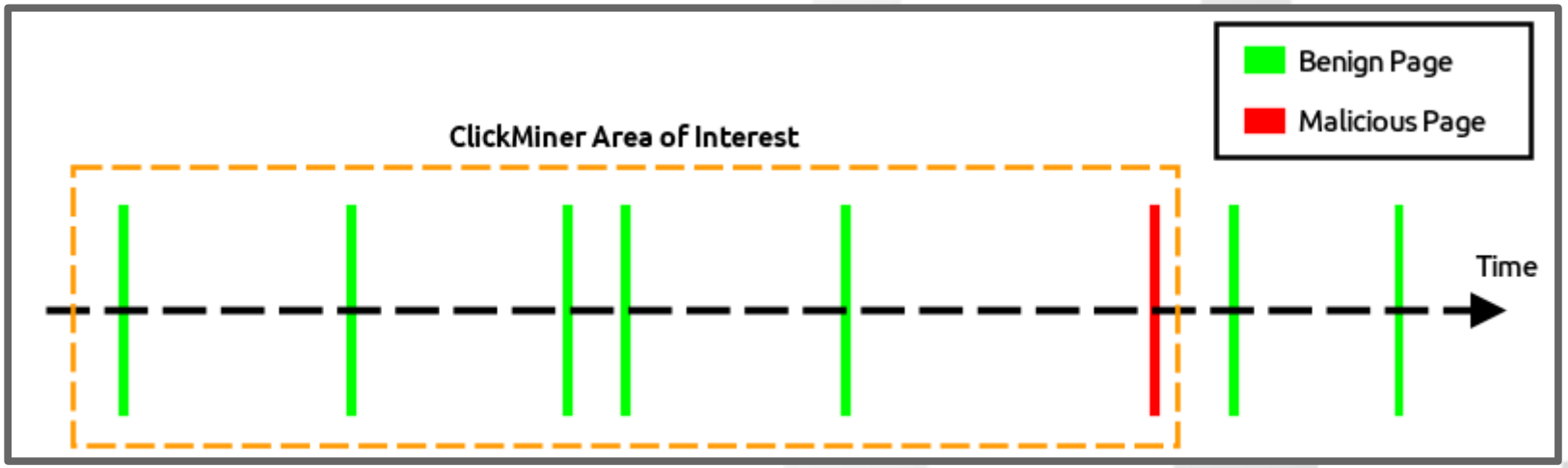
Requests: 6
Edges: 5
Clicks: 6
Time: 2m 56s



Network Security Intelligence

Interesting user-browser interactions

"What was a user's browsing behavior during a time window preceding (and including) a social engineering or phishing attack, or other relevant security incidents and anomalies?"



Security Network Intelligence

User-Browser Interaction

Click

- A user interaction that causes the browser to initiate an HTTP request for a new web page.
  - Mouse click on an image with an `onclick` event
  - Touch gesture on a form submit button
  - Pressing Enter while focused a link
  - Typing a URL into the address bar
  - Clicking on a bookmarked link

ClickMiner's Goals

- Accurately reconstruct the steps taken by the user to reach the attack page
- Filter out irrelevant requests

Previous Approach

Referrer Graph

- Node: HTTP request
- Edge: Defines request referrer → request referred relationship

ReSurf*

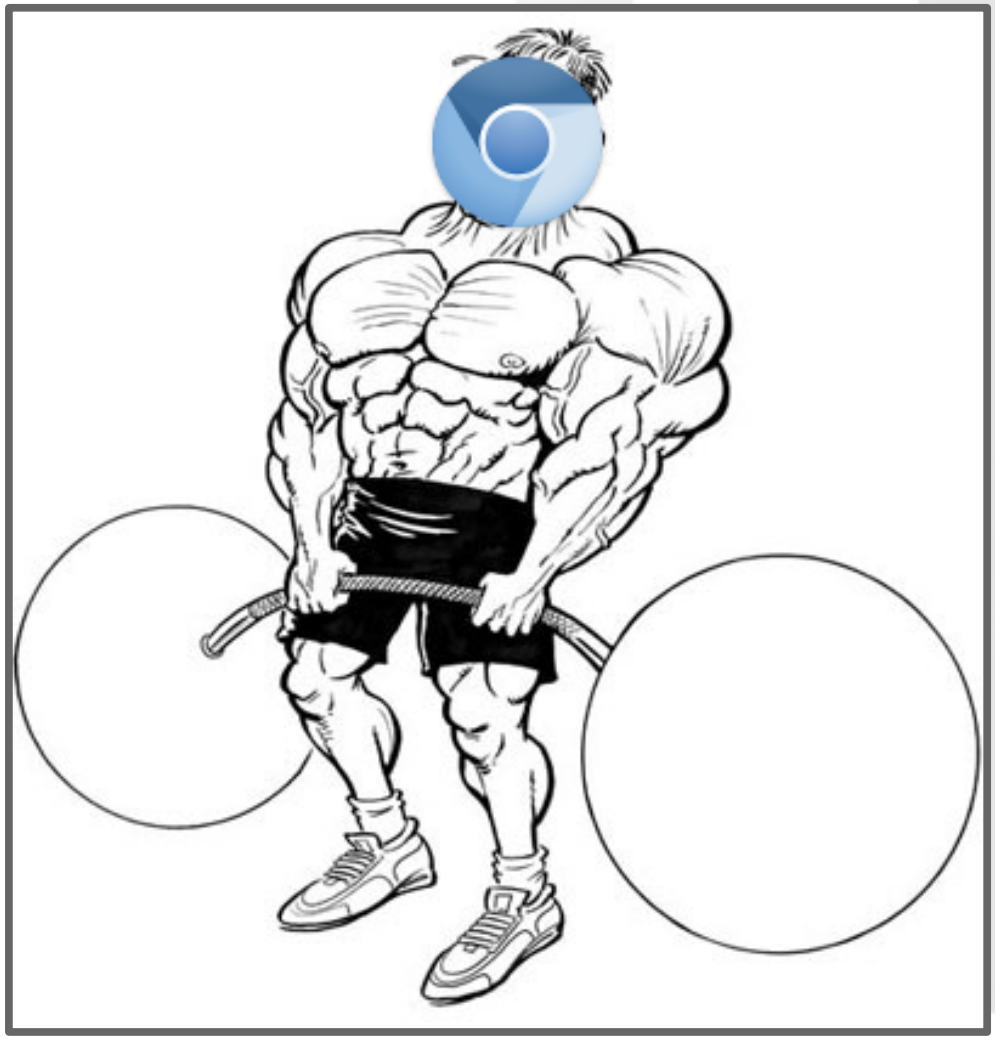
- Referrer-based click inference (*RCI*)
- Build *Referrer graph* from traffic
- Prune referrer graph based on heuristics
  - ex. Timing information between requests

*G. Xie et al. ReSurf: Reconstructing web-surfing activity from network traffic. In IFIP Networking Conference, 2013, 2013.

Security Network Intelligence

ClickMiner Approach

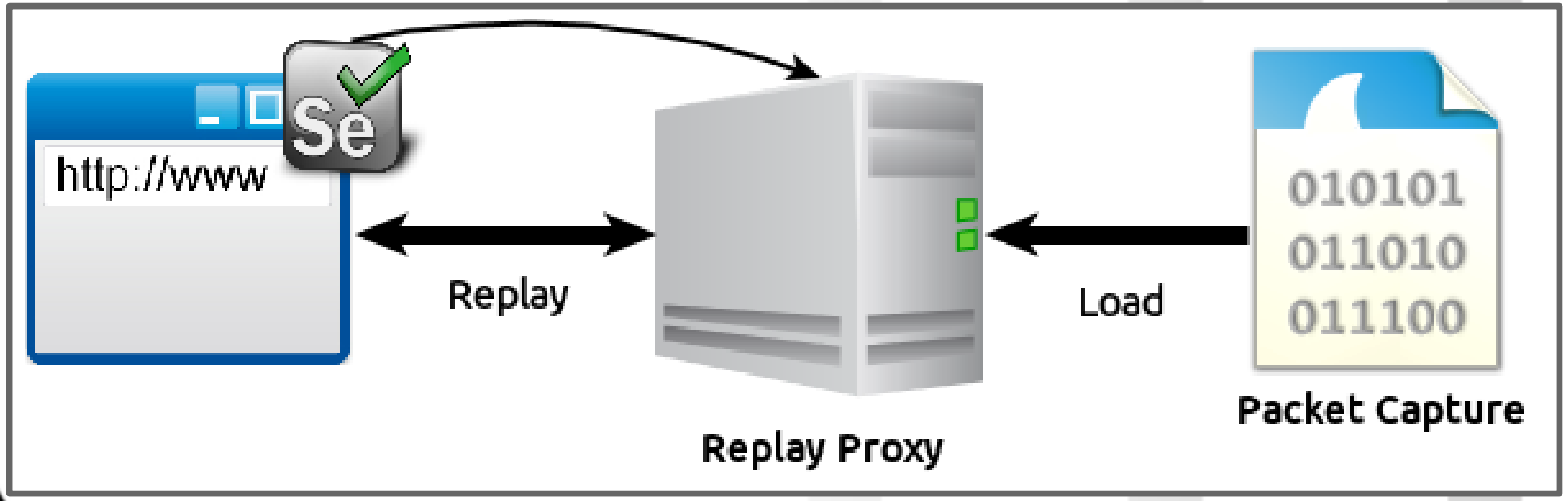
"Let the browser do the heavy lifting."



Network Security Intelligence

ClickMiner Approach

Replay web traffic within an instrumented browser.



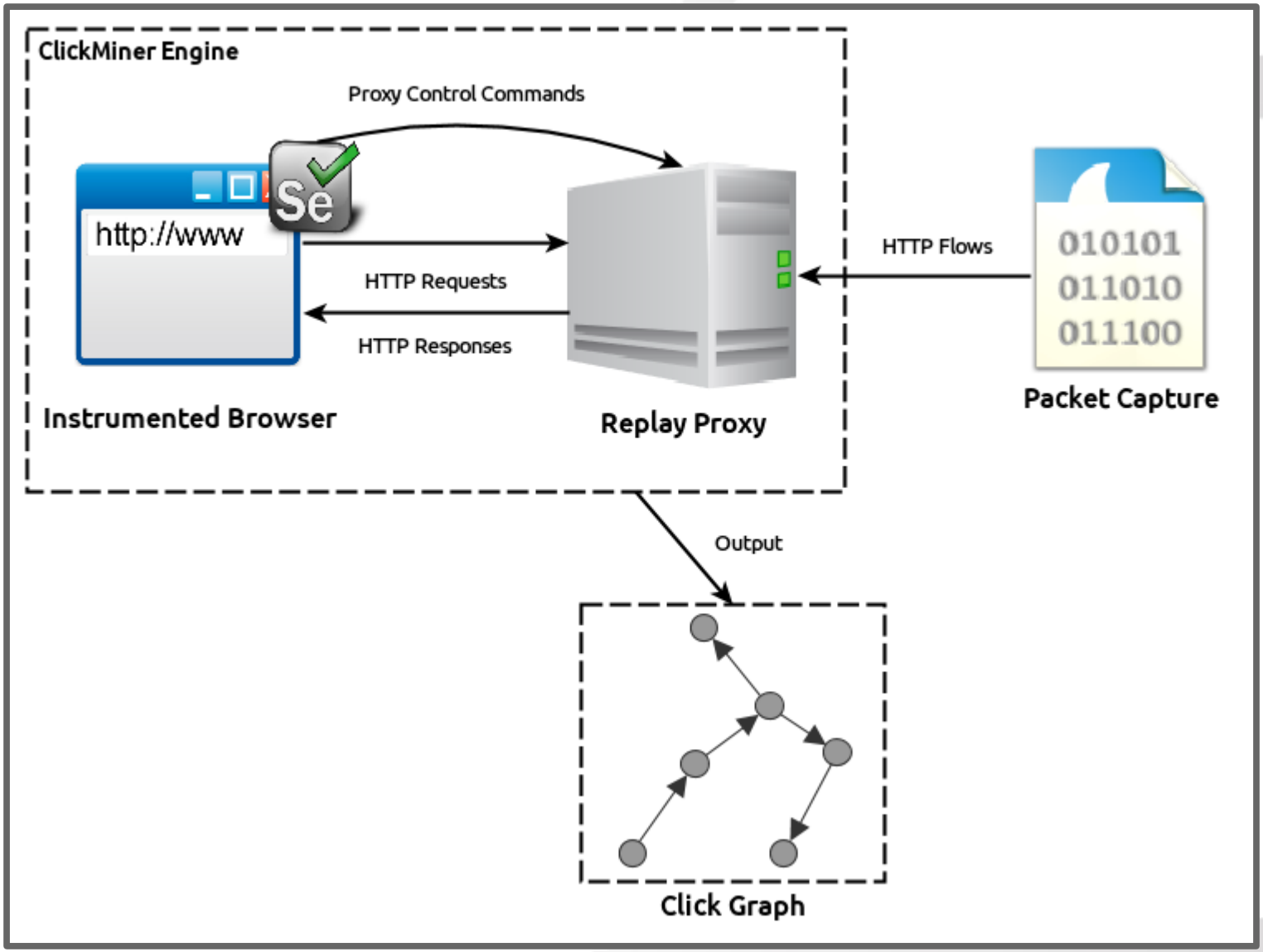
Security Network Intelligence

Contributions

- ClickMiner, a system dedicated to automatic reconstruction of clicks from web traffic.
- Evaluate both ClickMiner and RCI in a user study.
- Case study involving a real social engineering-based malware download attack.

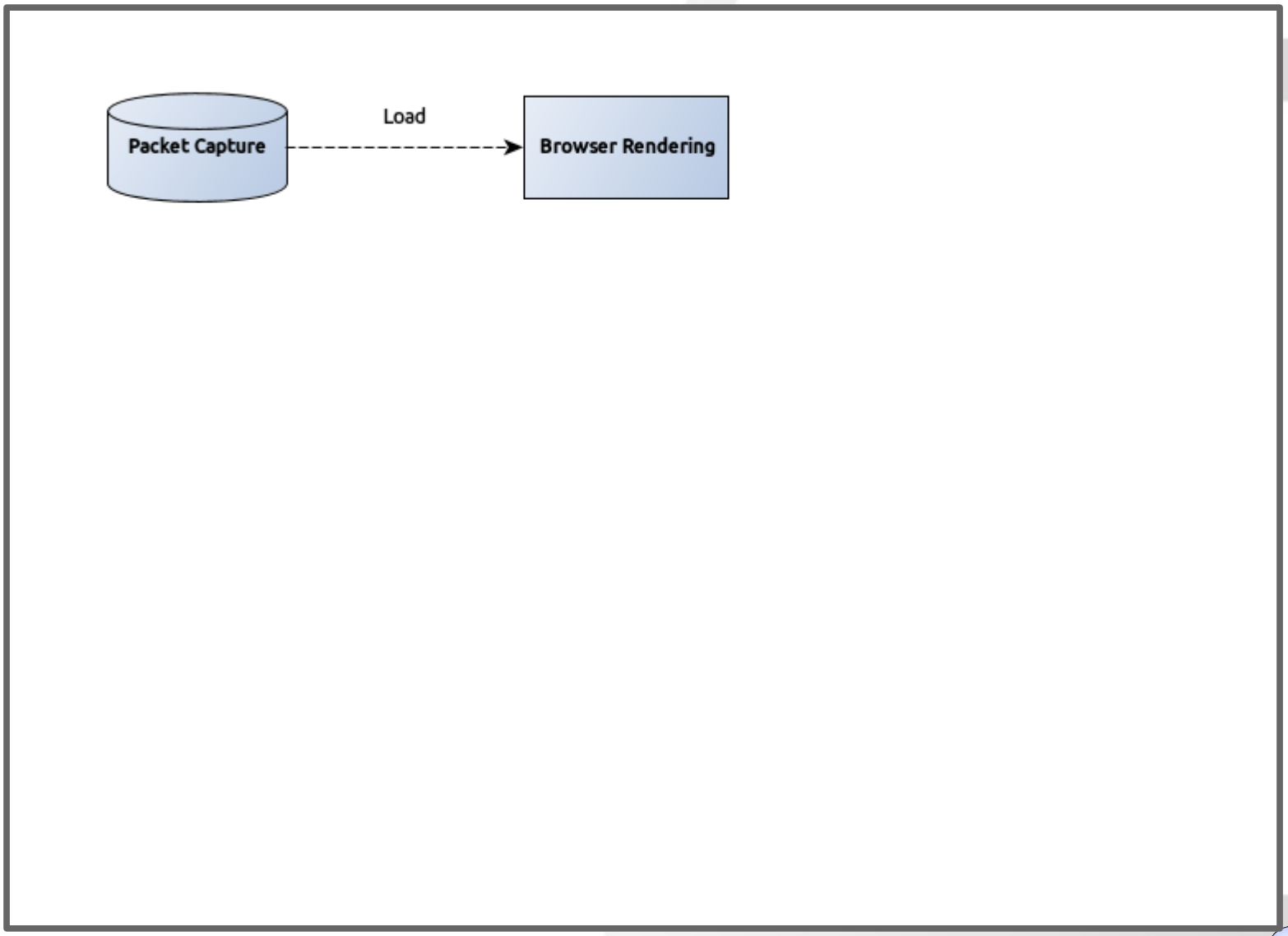
Network Security Intelligence

System Design

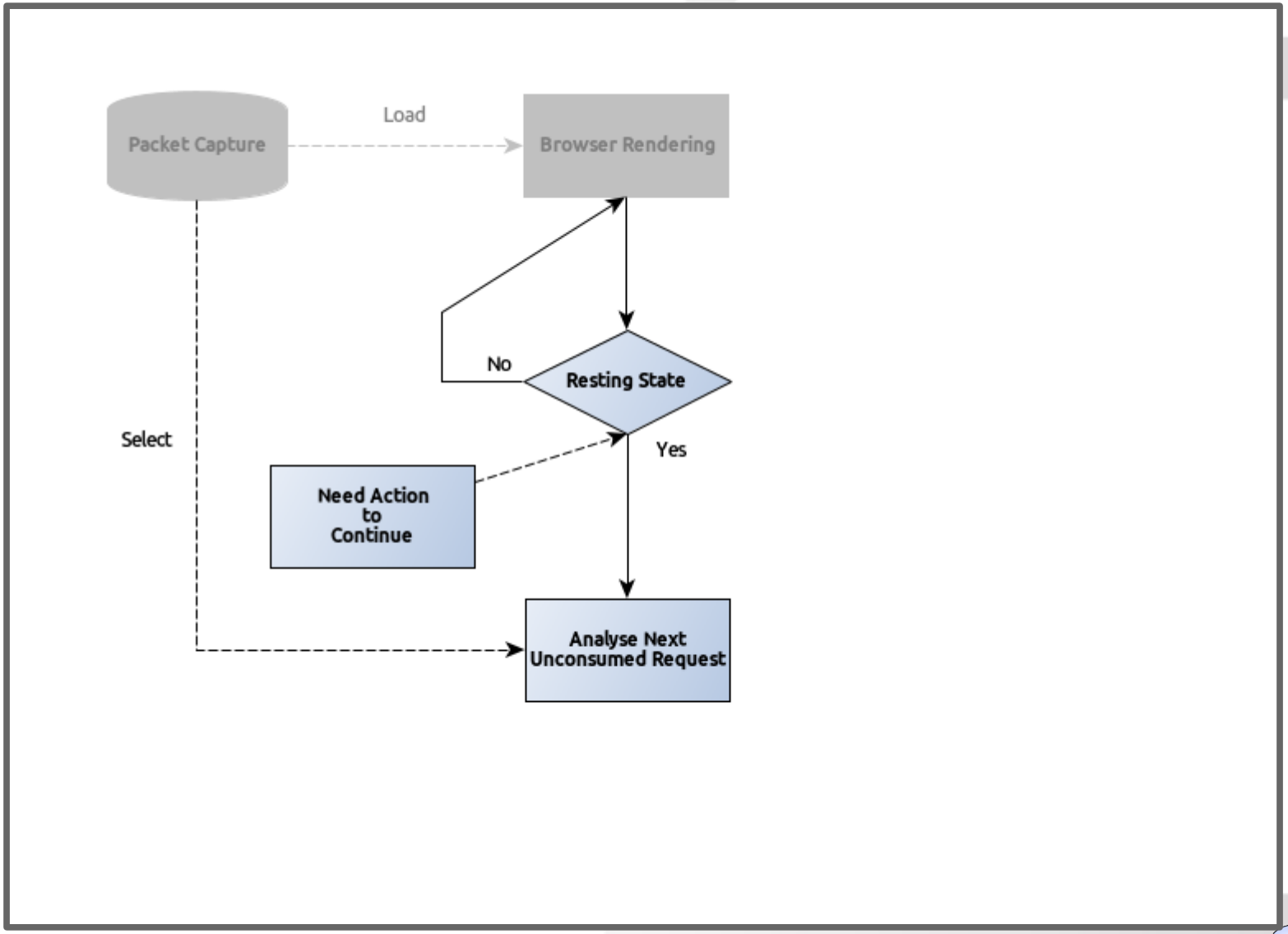


Network Security Intelligence

Replay Algorithm (Simplified)

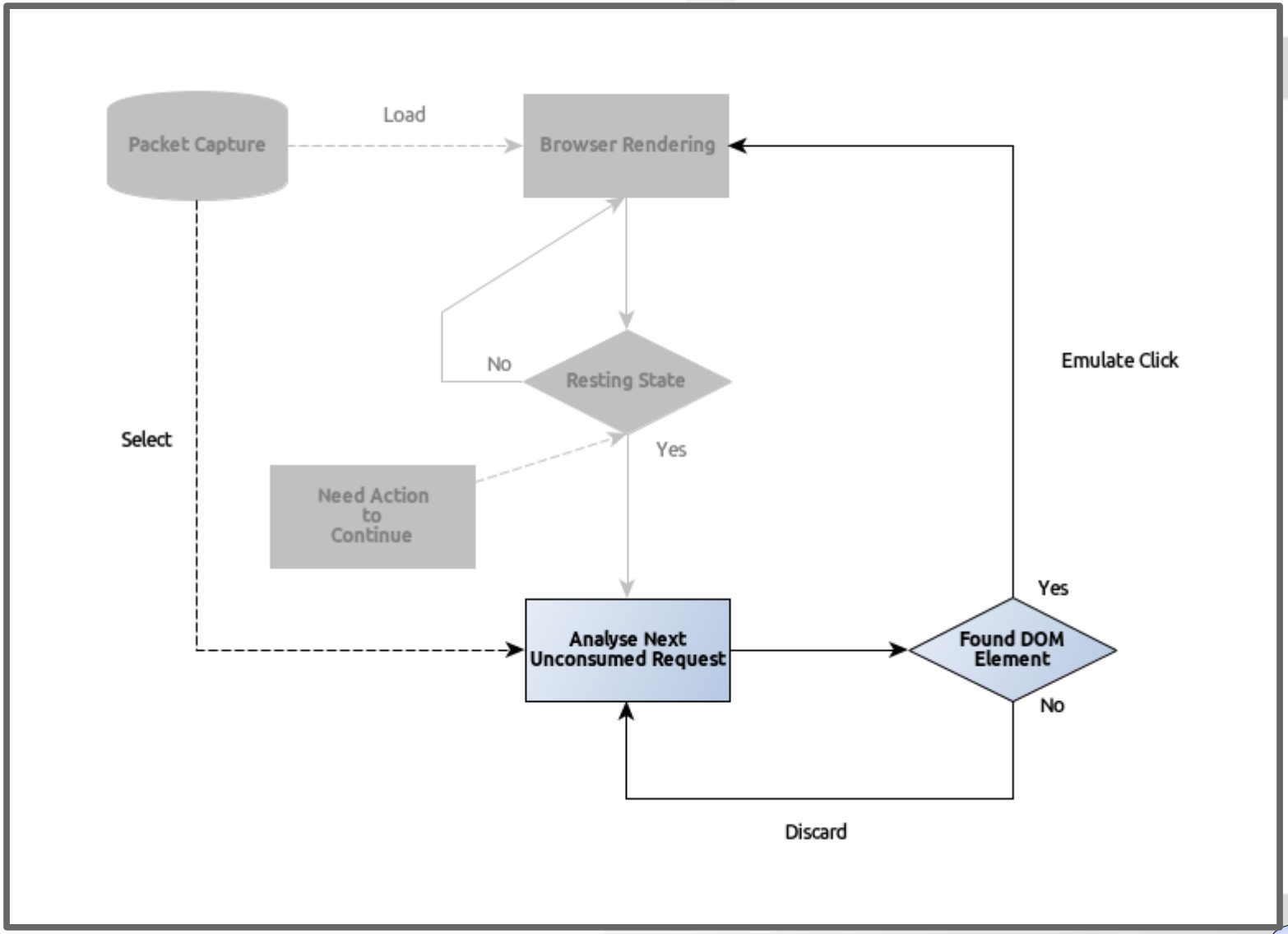


Replay Algorithm (Simplified)



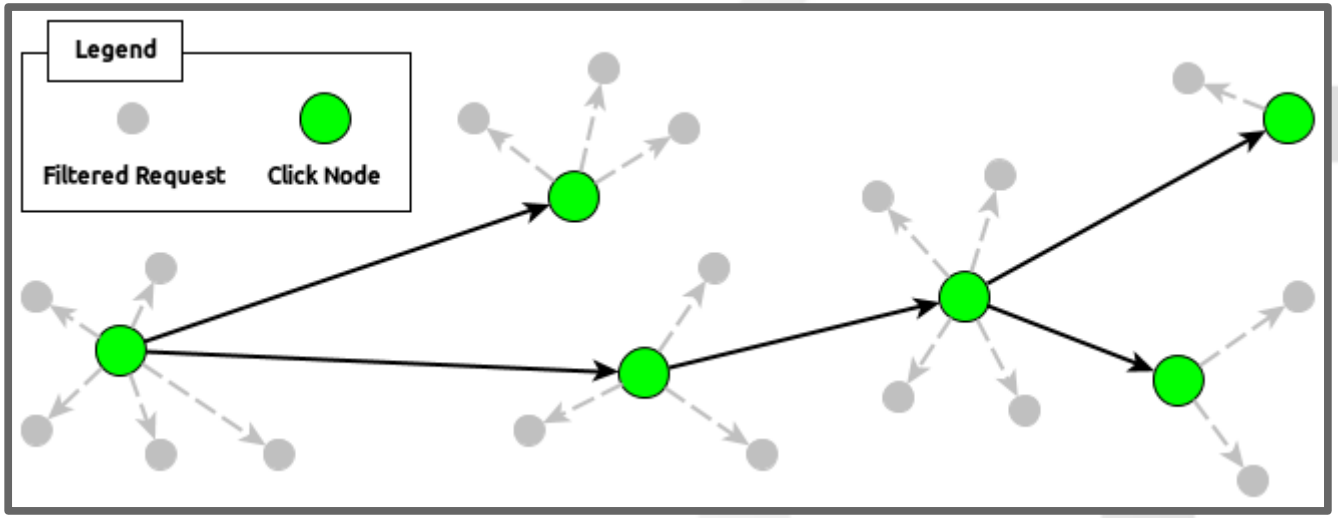
Network Security Intelligence

Replay Algorithm (Simplified)



Network Security Intelligence

Click Graph



Nodes: annotated HTTP Requests (p, e, q)

- p = source page
- e = element clicked
- q = request generated

Edge: $(p_w, e_w, q_w) \rightarrow (p_y, e_y, q_y)$

- p_y reached as a consequence of q_w

Security Network Intelligence

Additional Challenges

- Request URLs with dynamic content
- JavaScript mediated requests
- Browser Cache

Security Network Intelligence

Additional Challenges

Dynamically generated request can have different URLs between recording and replay

Request URLs with dynamic content

- URL parameter values
  - Randomly generated
  - Time-dependent
  - System-dependent
- Dynamically generated paths

Replay proxy utilizes an *approximate* matching algorithm for HTTP requests

Additional Challenges

Approximate matching algorithm compares HTTP requests based on:

```
http://sample.com/a/c?var1=a1&var2=b2
http://sample.com/a/c?var1=a1&var3=c3

Domain Path Parameters Values
```

If a match is found its response is served otherwise respond with HTTP `404`.

Network Security Intelligence

Additional Challenges

JavaScript Mediated Clicks

- DOM elements with JavaScript event handlers

Network-oriented best effort approach

- Discover JavaScript mediated elements
- Activate each one
- If expected HTTP request is generated then we've found the element
  - Otherwise respond with HTTP 204

Additional Challenges

Browser Cache

- Requests satisfied by the browser cache exhibit no response payload
- Requests with missing response payloads can not be replayed.
- *Best effort* replay skips gaps to continue processing what traffic remains.

Augmented Click Inference

ClickMiner might fail to detect click via replay

- Leverage the referrer graph
- Fill in click paths with partial click nodes

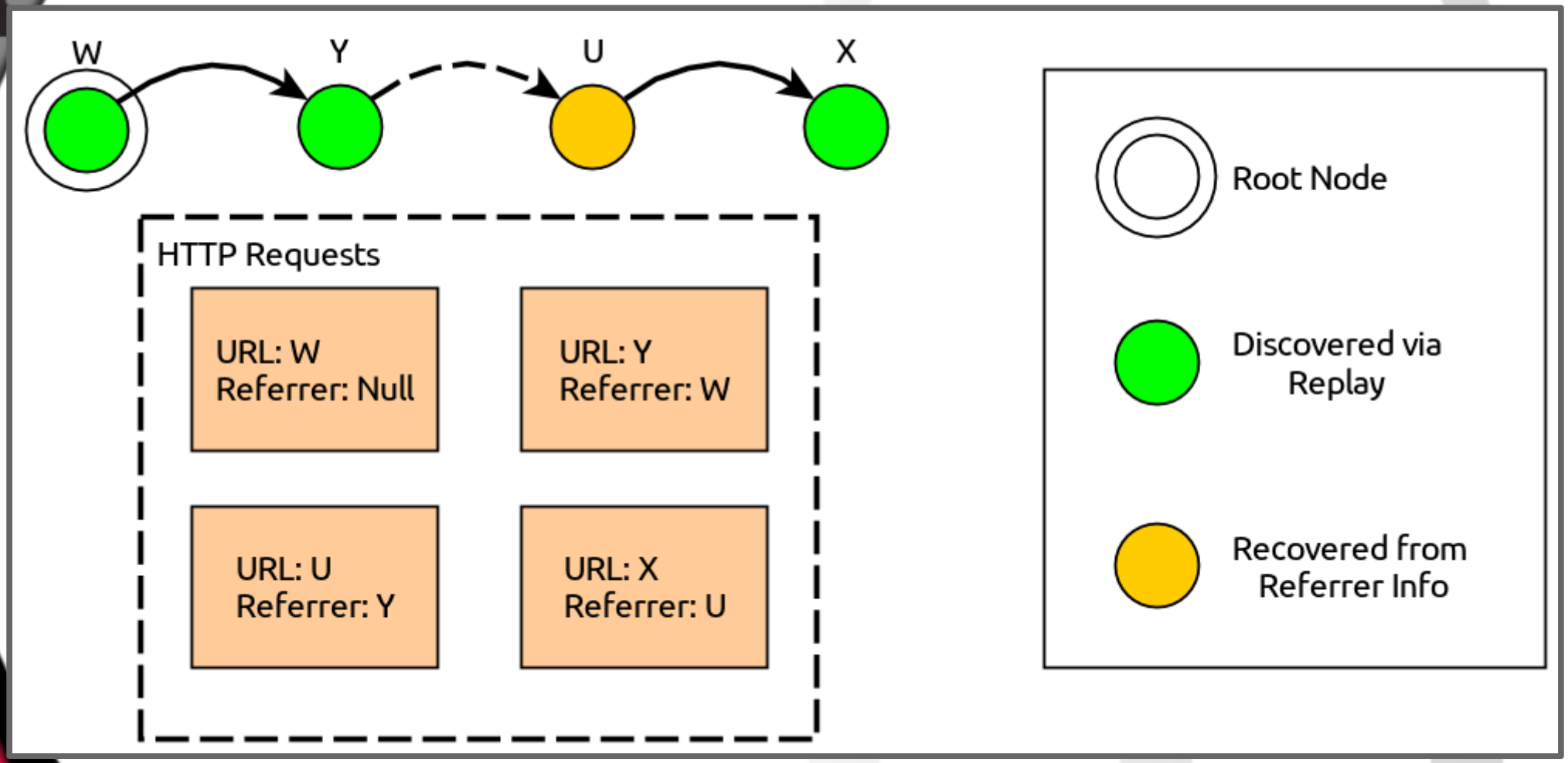
(***Click Graph***) + (***Referrer Graph***)

↓

Augmented Click Graph

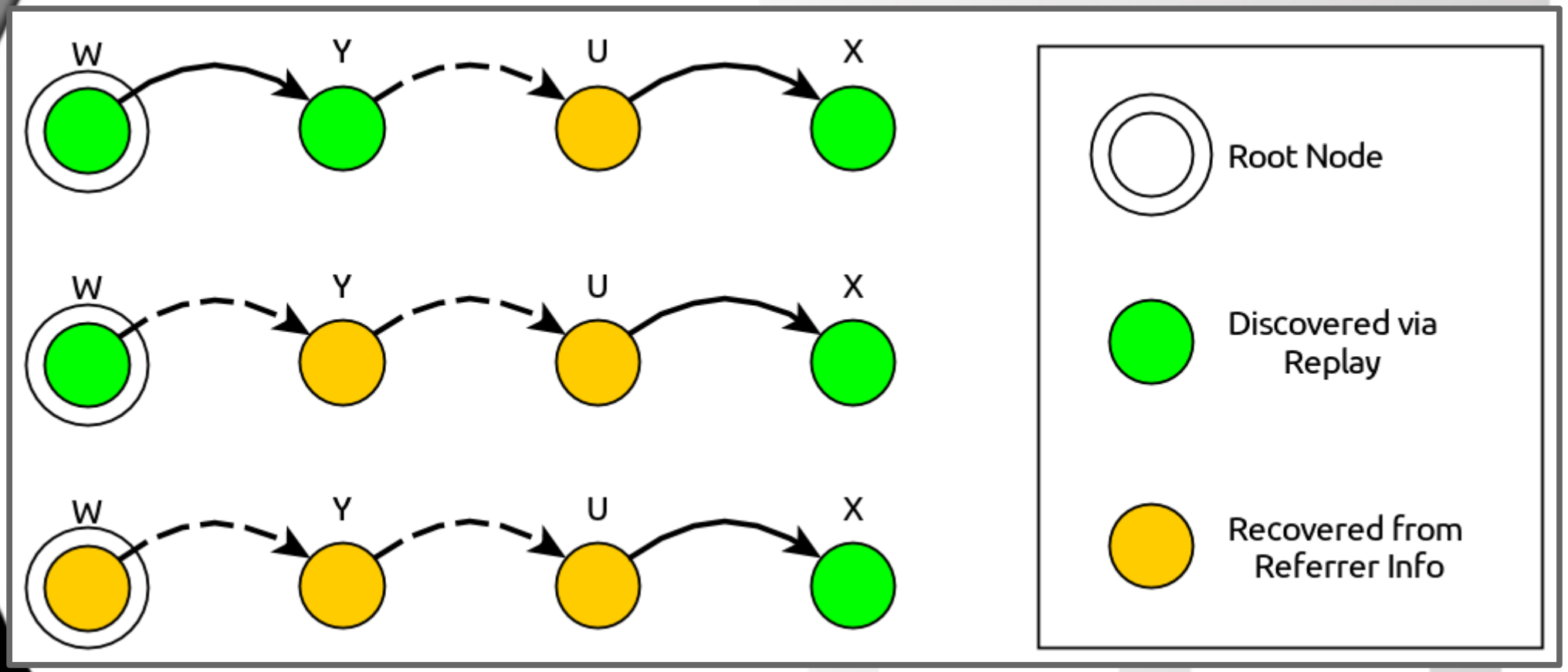
Network Security Intelligence

Augmented Click Inference



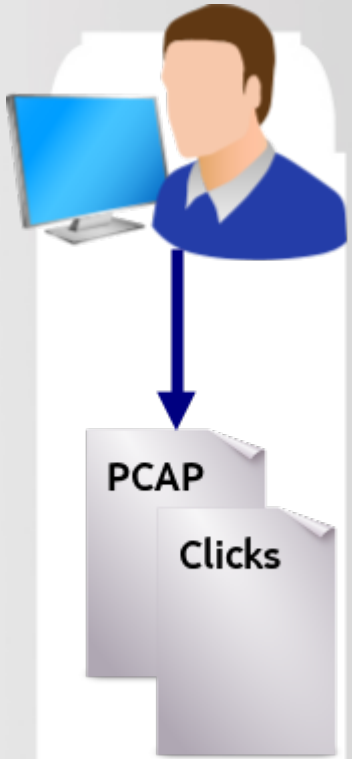
Network Security Intelligence

Augmented Click Inference



Evaluation

- Data Requirements
  - Clicks recorded at the browser level
    - Ground truth
  - Raw network traces
- User Study
  - Users performed generic web browsing activities
  - 21 Participants, 24 Traces
  - 2 Groups
    - Group 1: browser caching disabled
    - Group 2: browser caching enabled with “warmed up” cache



Network Security Intelligence

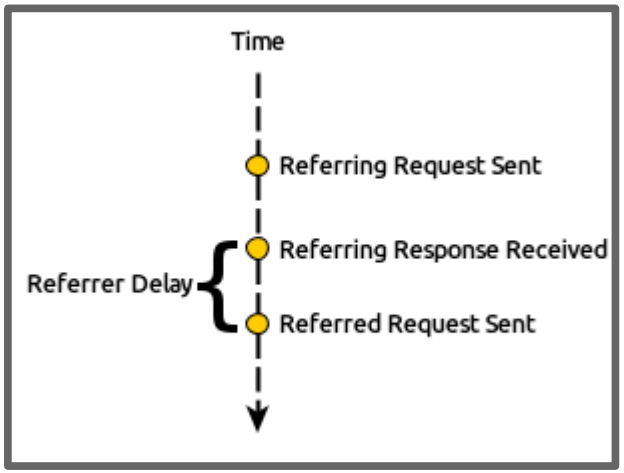
Results

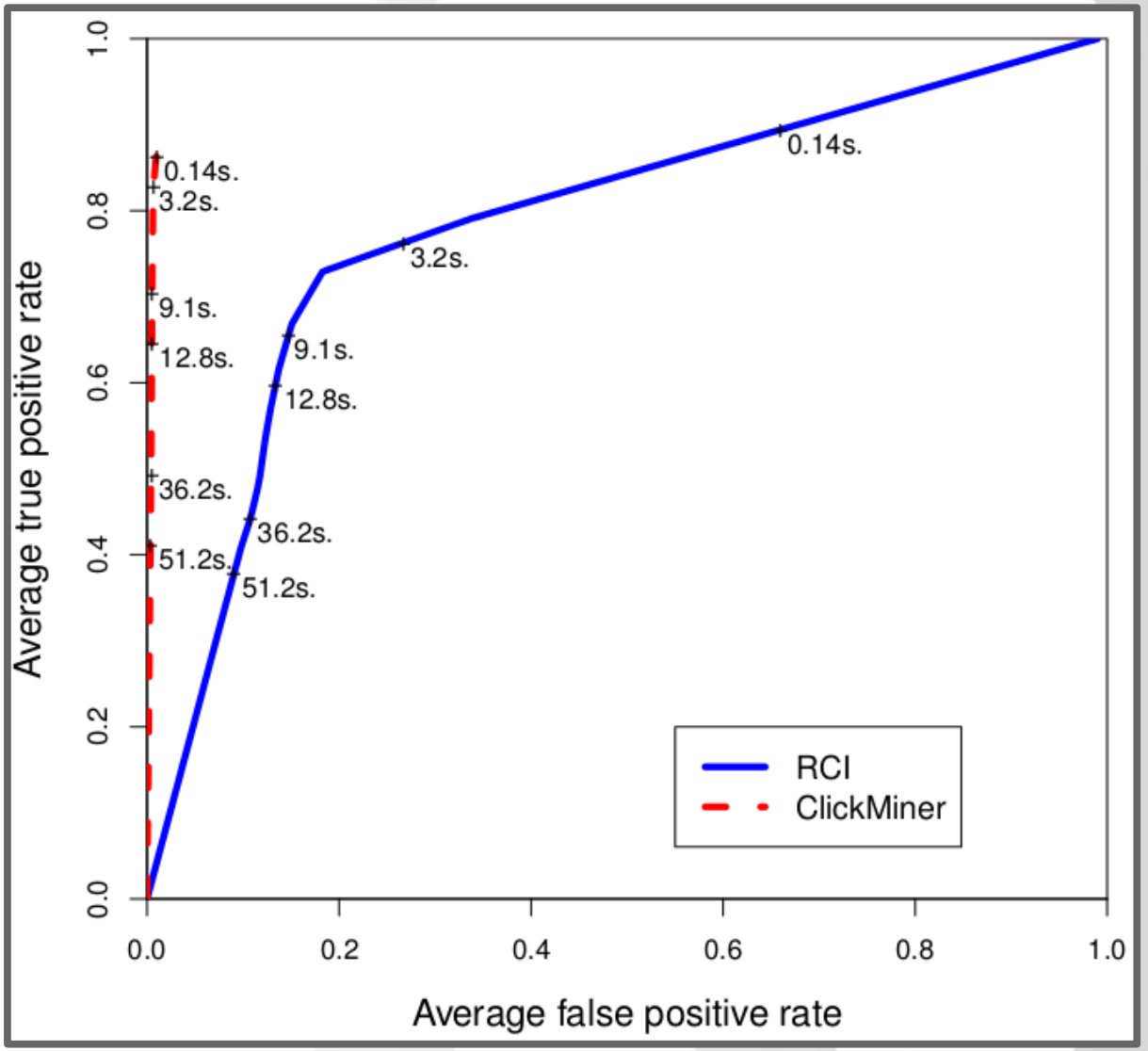
Summary

- Avg. between 82% and 90% of clicks correctly reconstructed
- Avg. between 0.74% and 1.16% false positives
- Greatly outperforms referrer-based approach

Network Security Intelligence

Results





Network Security Intelligence

Results

Trace Number	HTTP Requests	Recorded Clicks	Mined Clicks avg (stddev)	Matching Clicks avg (stddev)	TPR	FPR
1	3925	21	50.80 (0.40)	20.00 (0.00)	95.24%	0.79%
2	1114	25	39.00 (0.00)	25.00 (0.00)	100.00%	1.29%
3	2884	16	41.00 (0.00)	13.00 (0.00)	81.25%	0.98%
4	1030	10	16.00 (0.00)	10.00 (0.00)	100.00%	0.59%
5	3405	23	46.20 (0.75)	22.80 (0.40)	99.13%	0.69%
6	3800	21	51.60 (0.80)	19.00 (0.00)	90.48%	0.86%
7	4891	11	30.20 (0.40)	11.00 (0.00)	100.00%	0.39%
11	9247	37	75.00 (2.61)	32.20 (0.75)	87.03%	0.46%
14	6508	32	50.00 (1.10)	28.00 (0.00)	87.50%	0.34%
16	1167	32	28.60 (0.49)	22.00 (0.00)	68.75%	0.58%
18	4073	20	76.60 (1.50)	17.20 (0.40)	86.00%	1.47%
22	5005	23	51.40 (0.80)	21.00 (0.00)	91.30%	0.61%
23	722	14	15.00 (0.00)	11.00 (0.00)	78.57%	0.56%
Average	3674.69	21.92	43.95	19.40	89.63%	0.74%
Stddev	2350.46	7.88	18.21	6.60	9.58	0.34

Caching Disabled

Network Security Intelligence

Results

Trace Number	HTTP Requests	Recorded Clicks	Mined Clicks avg (stddev)	Matching Clicks avg (stddev)	TPR	FPR
8	4786	28	64.40 (0.80)	21.00 (0.00)	75.00%	0.91%
9	2212	19	42.80 (1.60)	14.00 (0.00)	73.68%	1.35%
10	1639	15	23.20 (0.40)	15.00 (0.00)	100.00%	0.50%
12	1219	10	15.60 (0.49)	7.00 (0.00)	70.00%	0.71%
13	1250	15	17.00 (0.00)	13.00 (0.00)	86.67%	0.32%
15	500	34	34.20 (0.40)	28.00 (0.00)	82.35%	1.33%
17	4682	25	63.00 (0.00)	19.00 (0.00)	76.00%	0.94%
19	2239	21	38.00 (1.26)	19.20 (0.40)	91.43%	0.85%
20	3980	21	117.00 (1.26)	19.00 (0.00)	90.48%	2.48%
21	2312	18	60.60 (0.49)	16.00 (0.00)	88.89%	1.93%
24	943	22	28.40 (0.49)	14.40 (0.49)	65.45%	1.52%
Average	2342.00	20.73	45.84	16.87	81.81%	1.16%
Stddev	1428.86	6.33	28.11	5.10	10.61	0.64

Caching Enabled

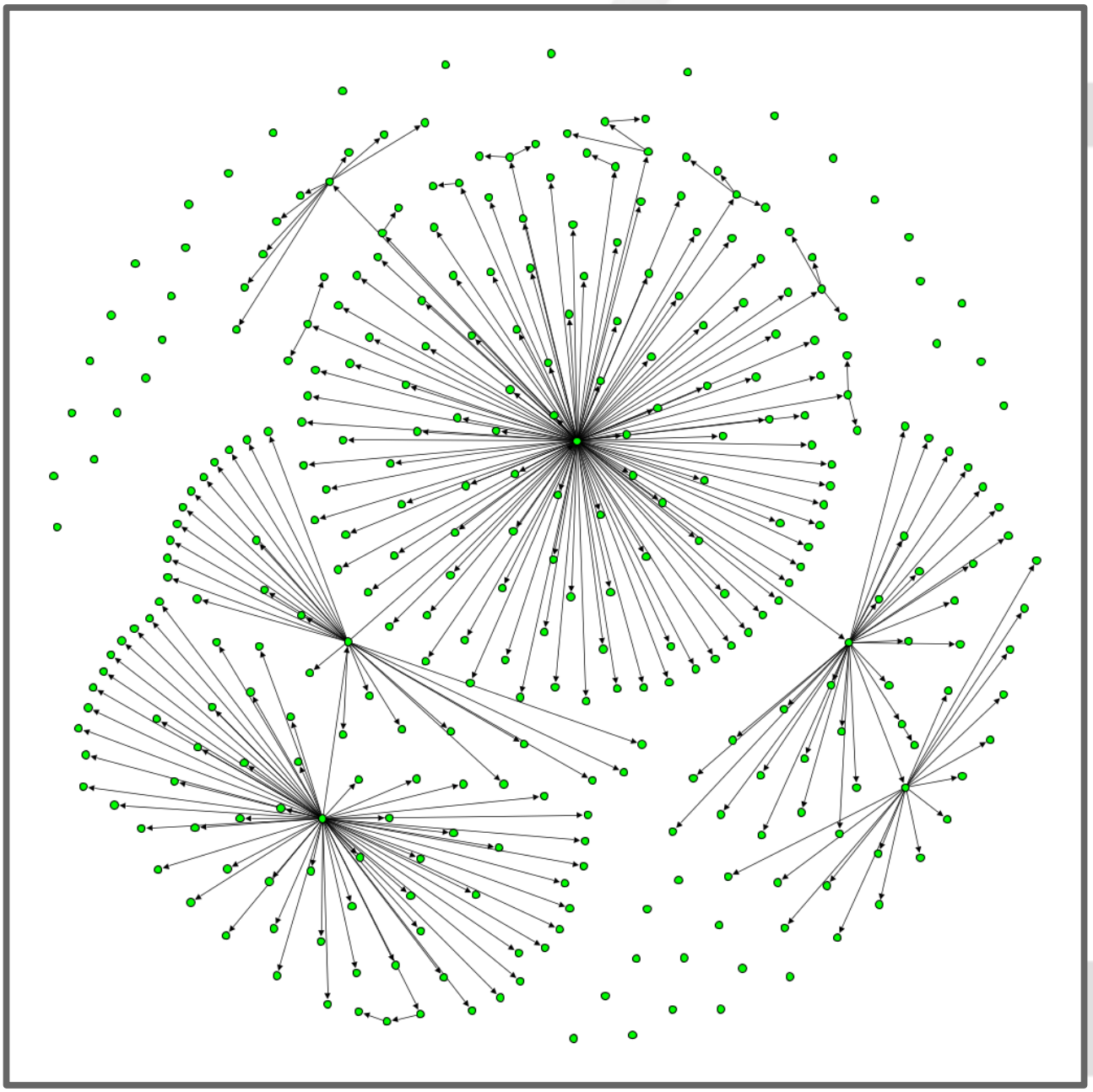
Network Security Intelligence

Case Study

Malware download incident from user study

- User visited `bing.com`
- User searched with term “far cry 3 hackz tools crack”
- User clicked on `allhackz[dot]net` from search results
- User clicked on “Download” button, opened two pages
  - `gameadvert[dot]com`
  - `wellmediaonline[dot]com`
- From `wellmediaonline[dot]com` download started via script from `effortlessdownload[dot]com`

Case Study

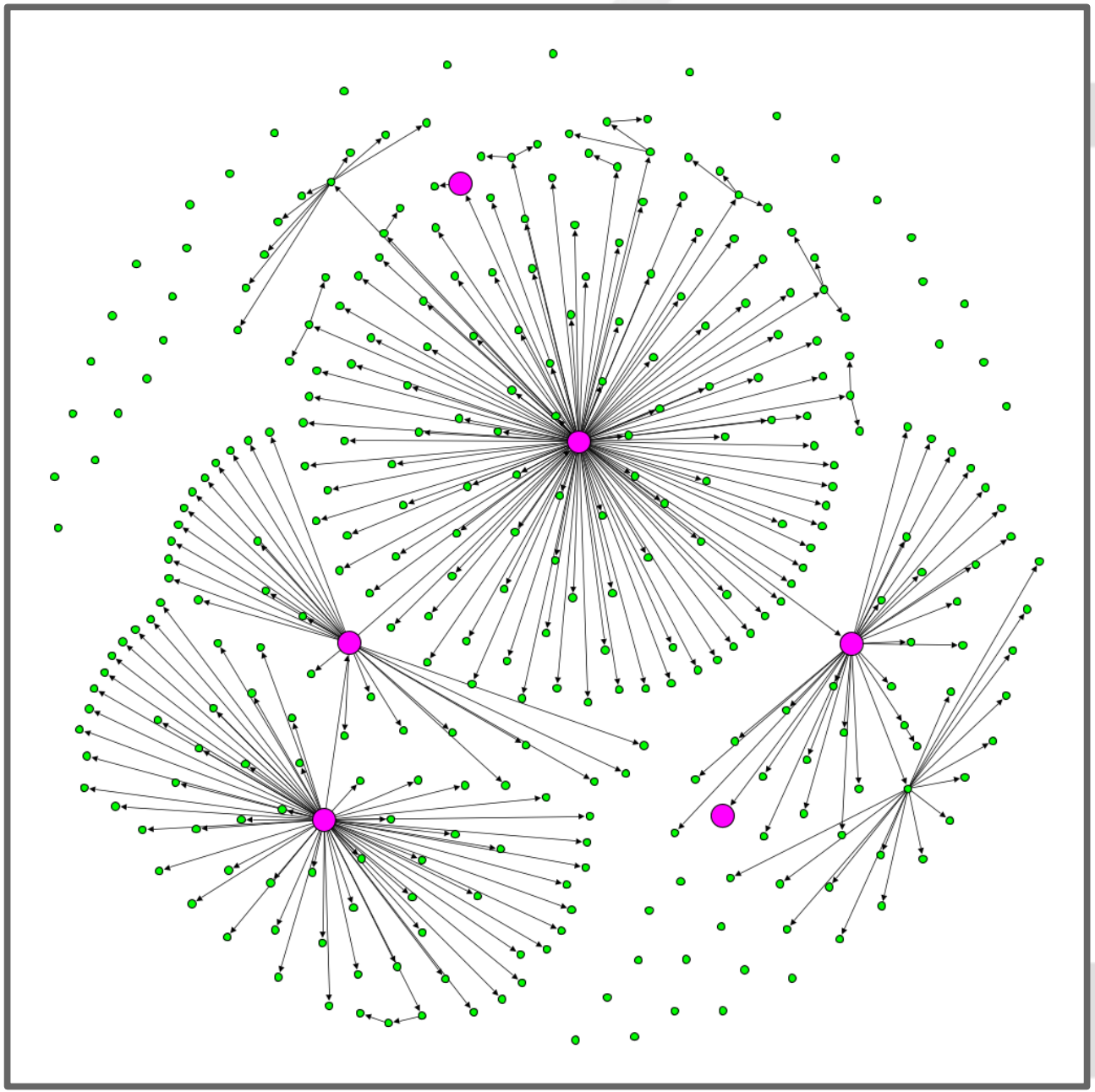


Requests: 328
Edges: 287
Clicks: 6
Time: 2m 56s

Network Security Intelligence

Case Study

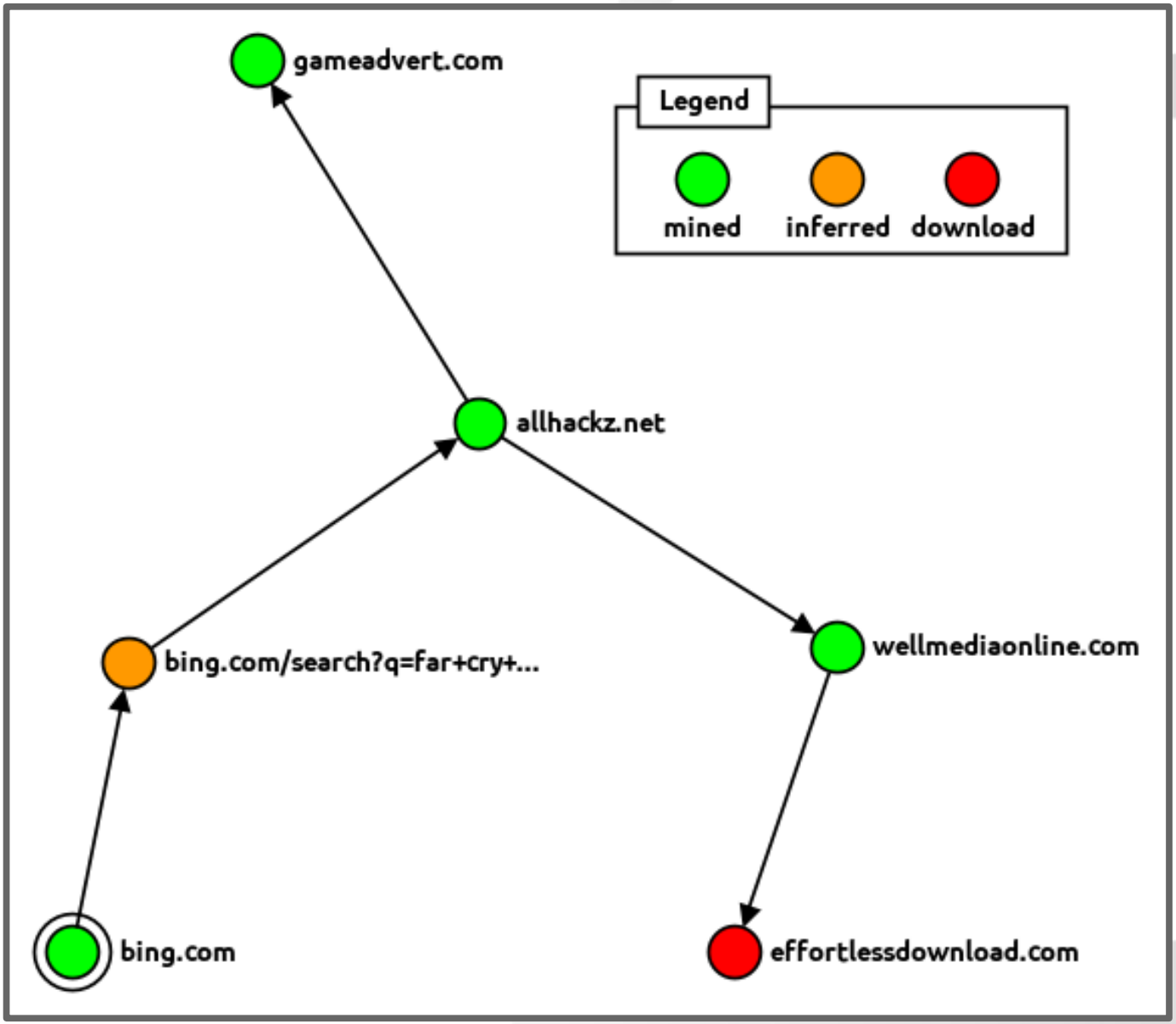
Requests: 328
Edges: 287
Clicks: 6
Time: 2m 56s



Network Security Intelligence

Case Study

Requests: 6
Edges: 5
Clicks: 6
Time: 2m 56s



Security Network Intelligence

Conclusion

- Importance of aiding the forensic analysis of web traffic traces
- ClickMiner, reconstructs user-browser interactions from network traces
- Through a user study we demonstrate:
  - Correctly reconstruct between a 82% and 90% of clicks
  - Low false positives
  - Outperforms exclusive referrer-based approach

Network Security Intelligence