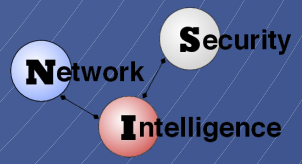
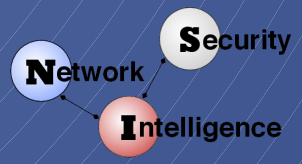
ClickMiner: Towards Reconstructing User-Browser Interactions from Network Traces

Chris Neasbitt
The University of Georgia
cjneasbi@uga.edu



Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study

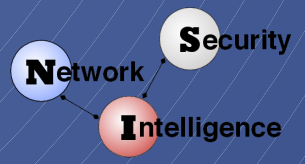


Problem

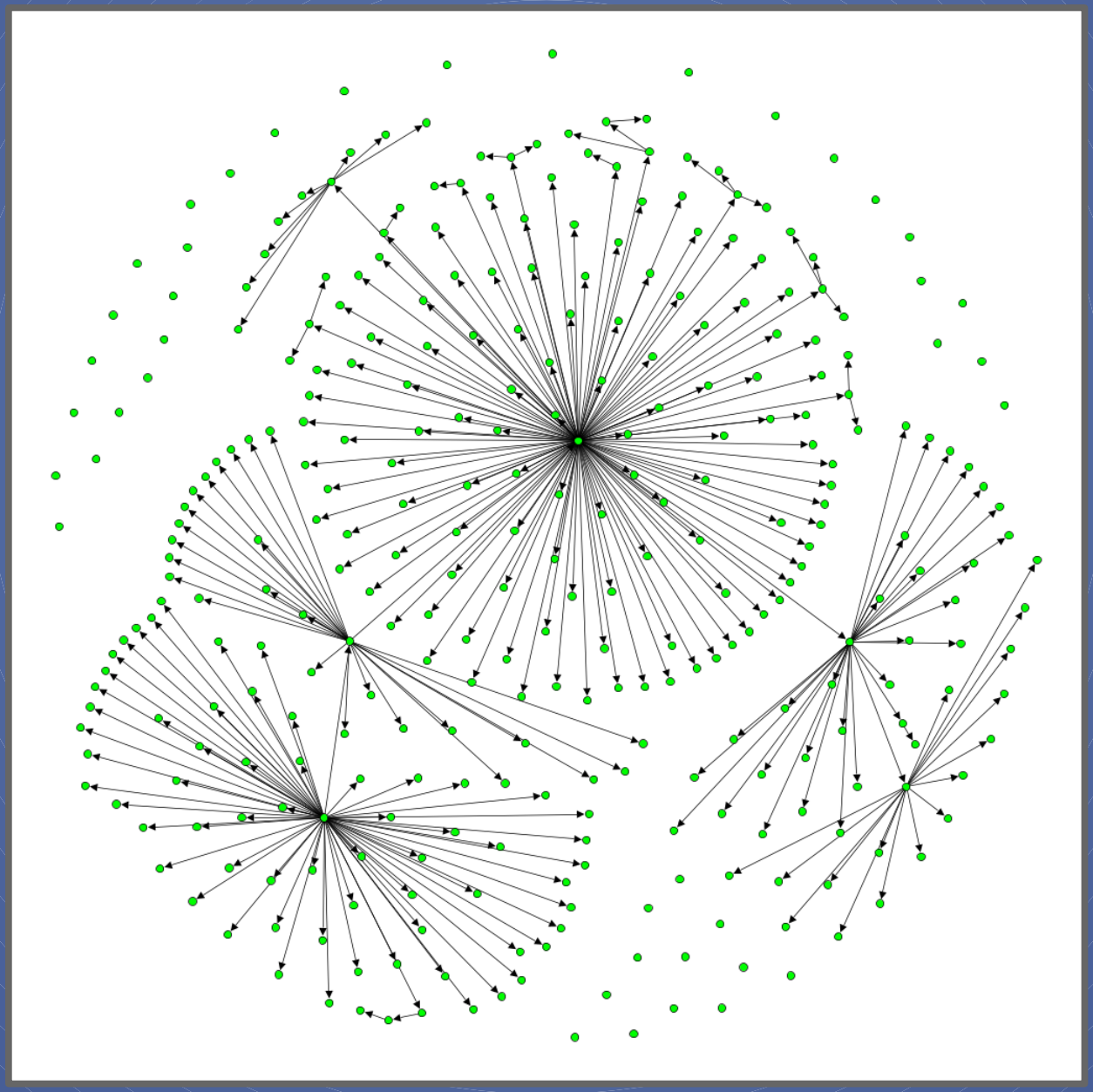
The modern web is becoming increasingly complex.

- Dynamic Pages
- Scripting Languages
  - e.g. JavaScript
- Browser Plug-ins
- Asynchronous requests

Increasing *Semantic Gap* between network traffic and user actions.



Problem

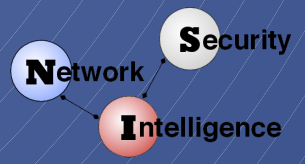


Problem

"Given the network traffic trace of a browsing session can we determine what interactions with the browser a user made?"

Benefactors

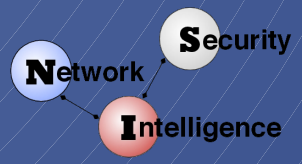
- Forensic Analysis
- Web Usage Miners



Problem

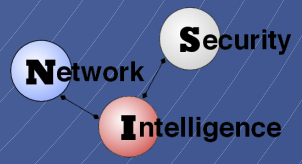
User-browser interaction i.e. *click*

- A user interaction that causes the browser to initiate and 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 to follow it
  - Typing a URL into the address bar
  - Clicking on a bookmarked link



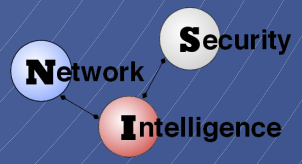
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



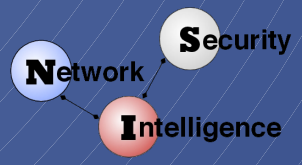
Goals

- Accurately infer *clicks* from full packet network traces.
- Reconstruct the sequence of web pages explicitly requested by the user.
- Infer what page element(s) in a web page was clicked by the user



Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



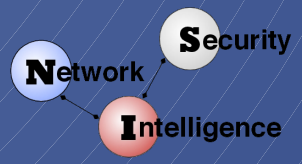
Previous Approach

ReSurf

- Referrer-based click inference (RCI)
- Build *Referrer graph* from traffic
- Prune referrer graph based on heuristics

Referrer Graph

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



ClickMiner Approach

"Let the browser do the heavy lifting."

1785

Network Security Intelligence

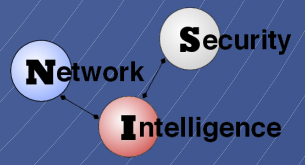
ClickMiner Approach

Network traffic replay within an instrumented browser.

- Through its execution the browser will *consume* traffic.
- Analyze what remains against open pages.

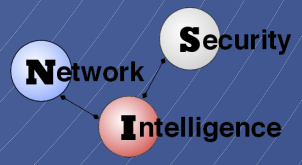
Click graph analysis of replay results.

- Utilize referrer information to fill in gaps



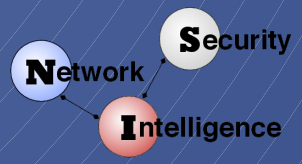
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



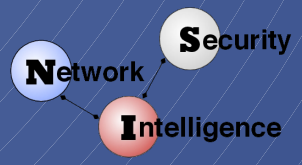
Contributions

- ClickMiner, a novel system dedicated to automatically reconstructing user-browser interactions from full packet captures.
- Evaluate both ClickMiner and RCI in a user study.
- Case study involving a real social engineering-based malware download attack.

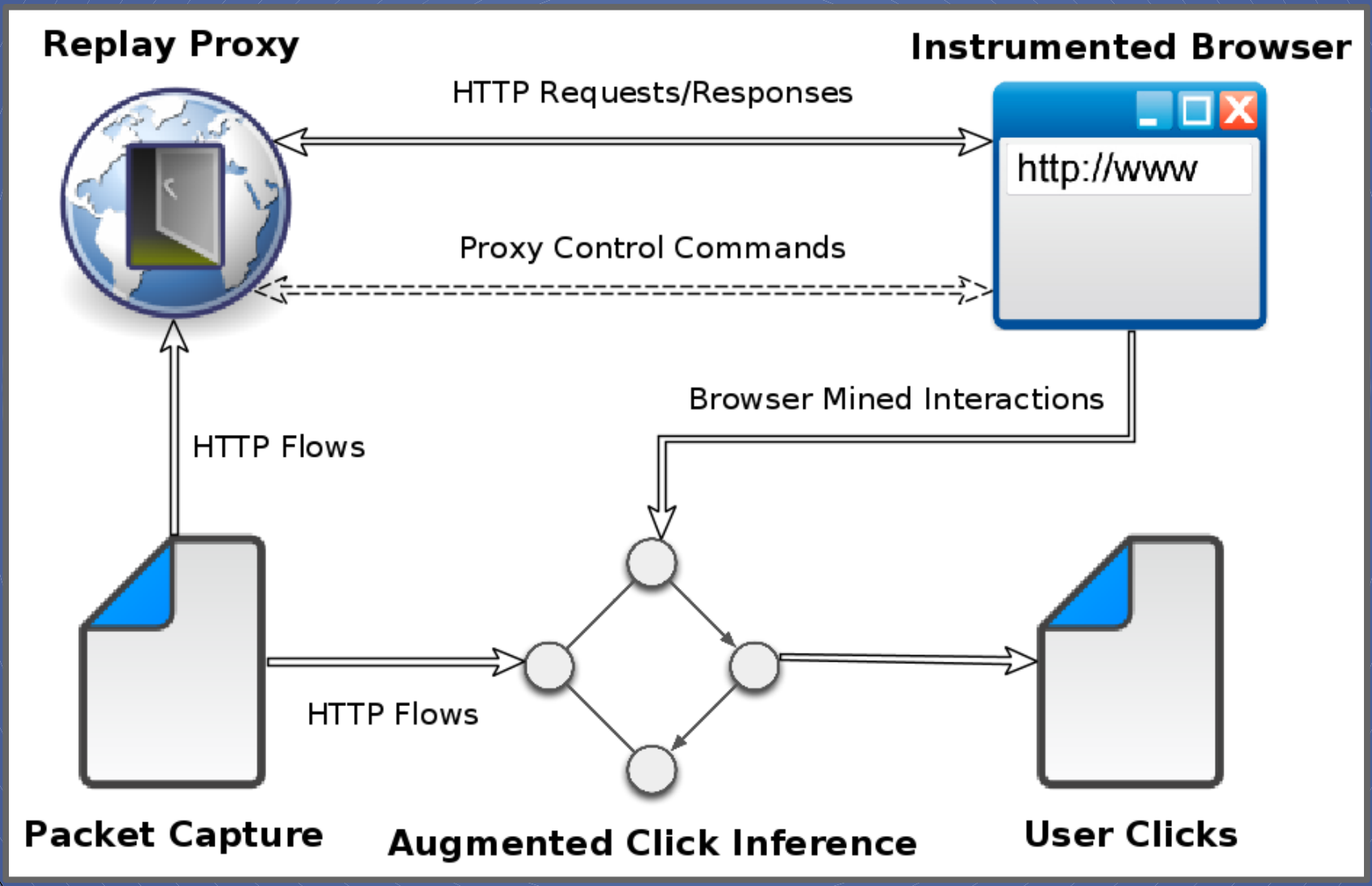


Outline

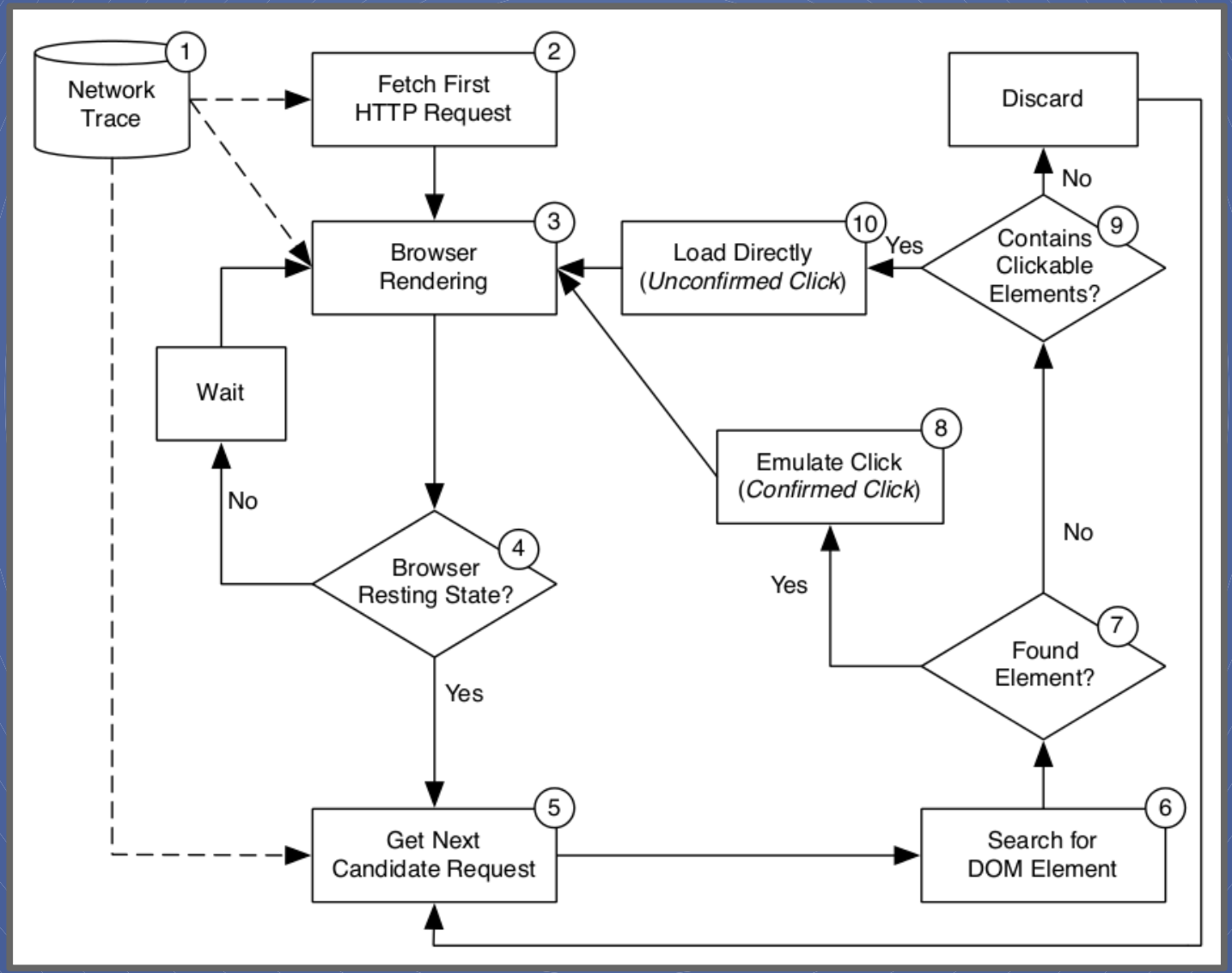
- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



System Design



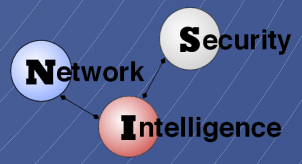
System Design



System Design

Click Graph

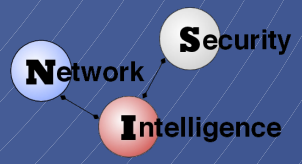
- Nodes: annotated HTTP Requests (p, e, q)
  - p = source page for the click
  - e = element clicked during interaction
  - q = HTTP request generated
- Edge: $(p_w, e_w, q_w) \rightarrow (p_y, e_y, q_y)$
  - p_y reached if as a consequence of q_w



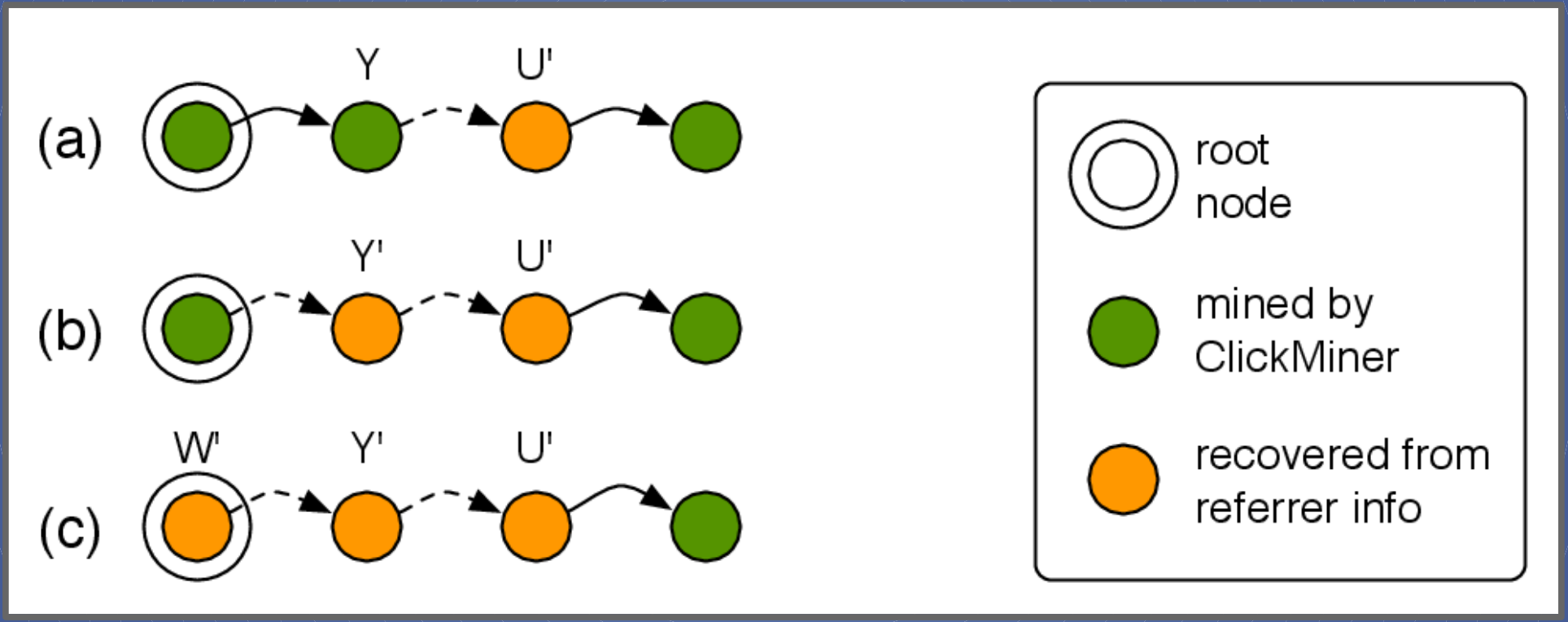
System Design

ACI (Augmented Click Inference)

- ClickMiner might fail to detect click.
- Leverage the referrer graph
- Fill in the gaps in click paths with partial click nodes

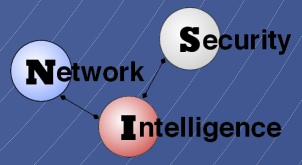


System Design



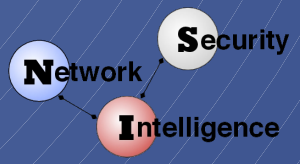
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



Challenges

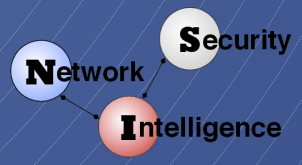
- Missing content
- Request URLs with dynamic content
- JavaScript mediated requests
- HTTPS



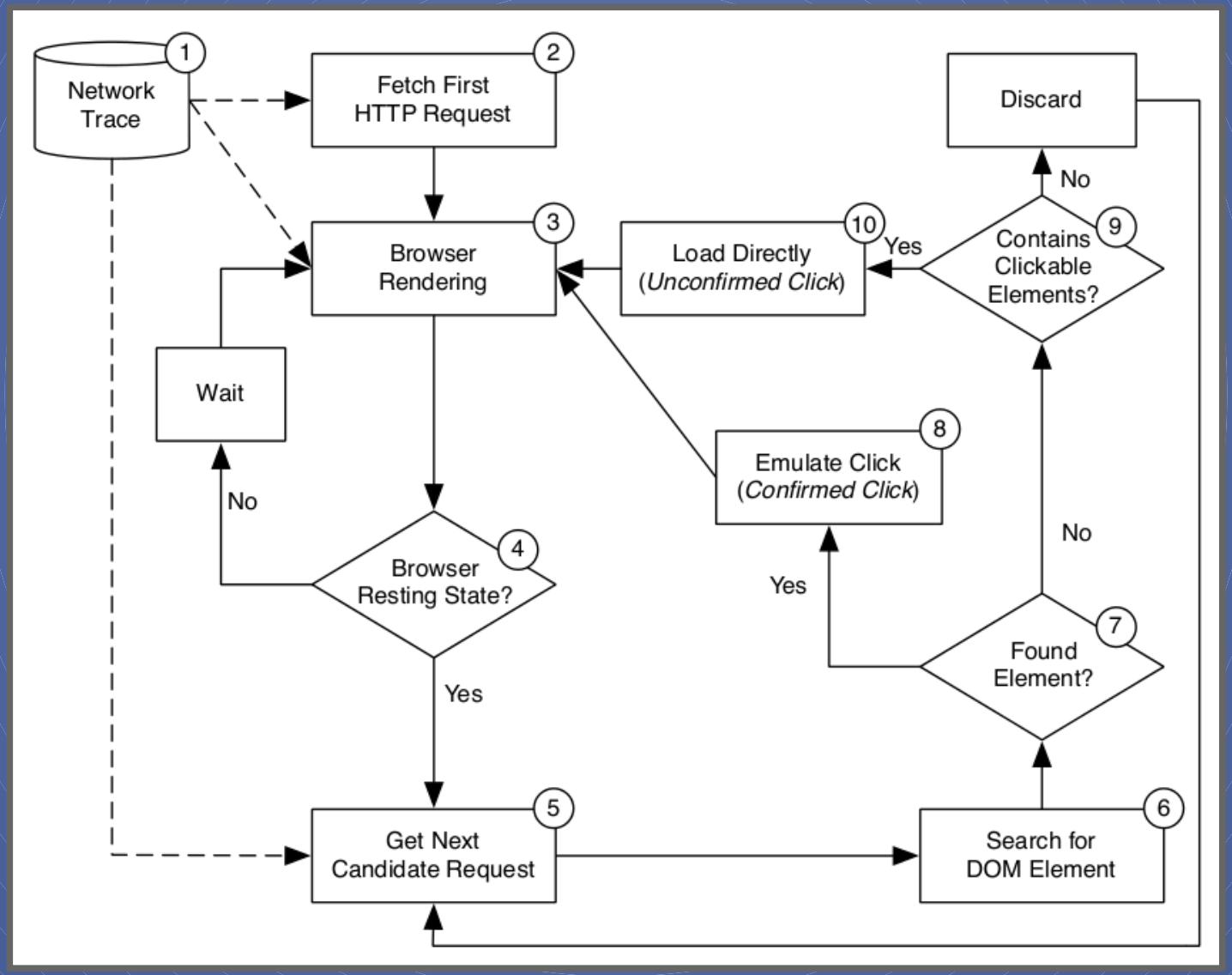
Challenges

Missing Content

- Requests with missing response payloads can not be replayed.
  - Browser Cache
  - Corrupted or Loss Packets
- *Best effort* replay skips these gaps to continue processing what traffic remains.



Challenges

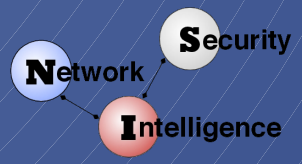

1
Network Trace
2
Fetch First HTTP Request
3
Browser Rendering
4
Browser Resting State?
No
Wait
5
Get Next Candidate Request
6
Search for DOM Element
7
Found Element?
Yes
No
8
Emulate Click (Confirmed Click)
Yes
9
Contains Clickable Elements?
No
Discard
10
Load Directly (Unconfirmed Click)
Yes

Challenges

Request URLs with dynamic content

- URL parameters containing:
  - Randomly generated values
  - Time-dependent values
  - System-dependent values
- Dynamically generated paths

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

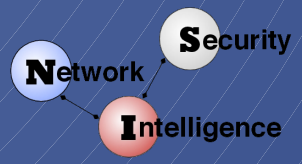


Challenges

Approximate matching algorithm compares HTTP requests based on:

- Domain name or IP address
- URL path
- URL parameter names
- URL parameter values
- Timestamps

If a match is found it's response is served otherwise respond with 404.



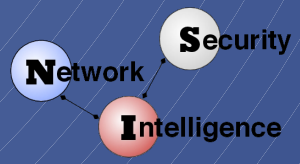
Challenges

JavaScript Mediated Clicks

- DOM elements with JavaScript event handlers

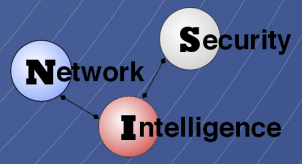
Network-oriented approach

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



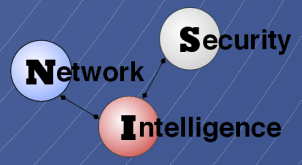
Challenges

- HTTPS
  - Migration toward ubiquitous use on the web
  - Many enterprise networks already deploy SSL-MITM proxies
    - mitmproxy
    - HoneyProxy
    - Paros
    - Burp



Outline

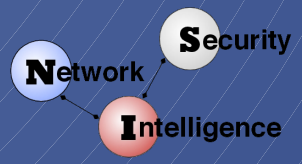
- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



Evaluation

User Study

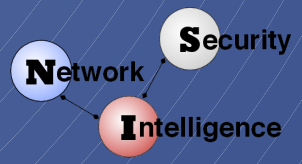
- Users performed generic web browsing activities
- Both traffic trace and user interactions were recorded
- 21 Participants, 24 Traces
- 2 Groups
  - Group 1: browser caching disabled
  - Group 2: browser caching enabled with “warmed up” cache



Evaluation

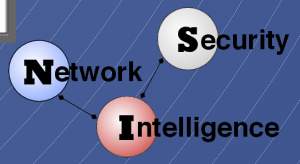
ClickMiner Results Summary

- Avg. between 82% and 90% of clicks reconstructed
- Avg. Between 0.74% and 1.16% false positives
- Greatly outperforms RCI



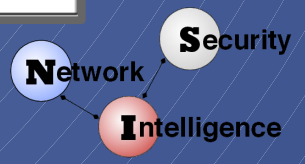
Evaluation

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

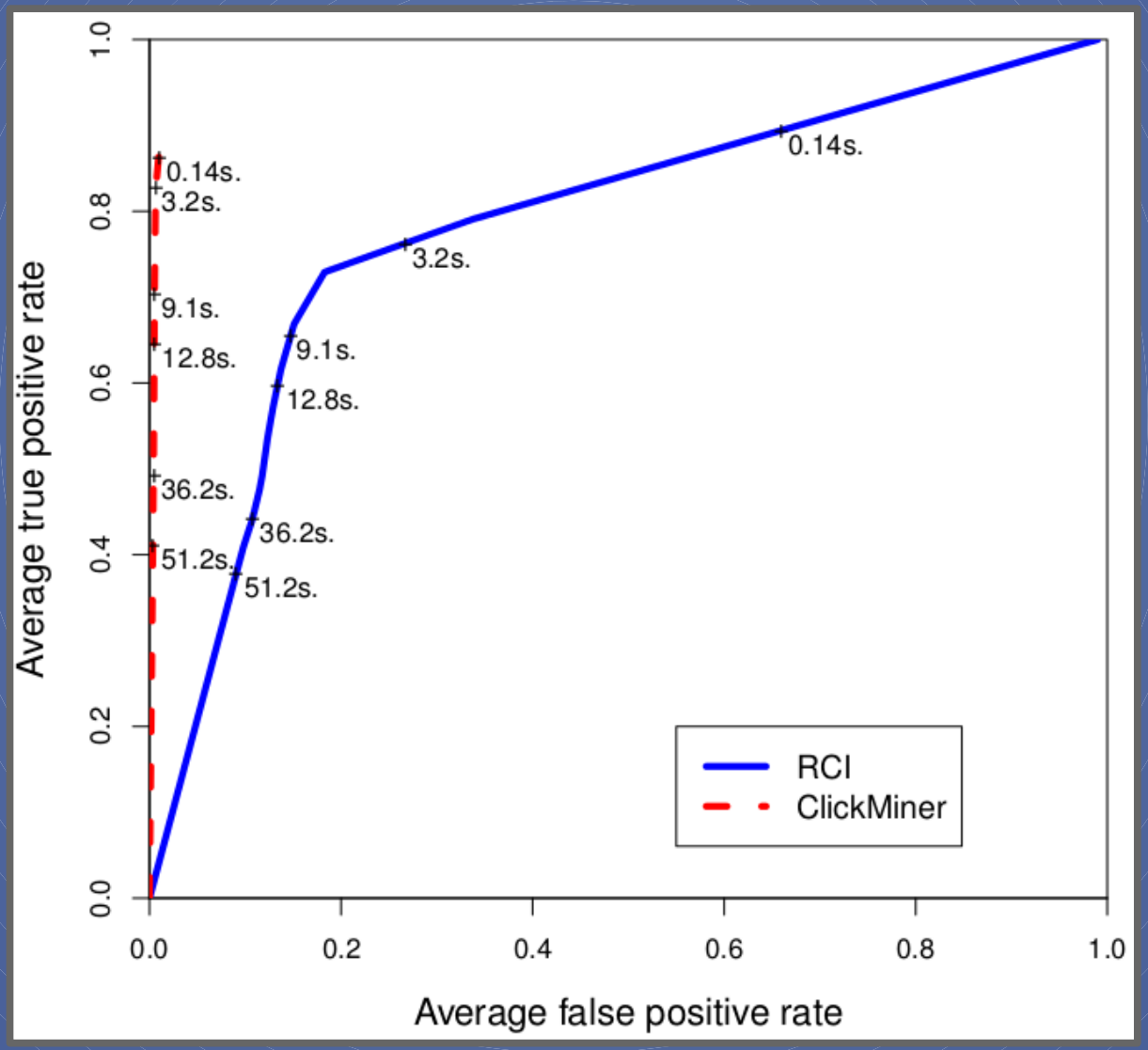


Evaluation

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

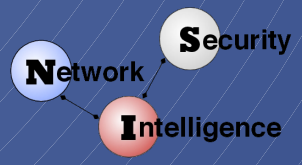


Evaluation



Outline

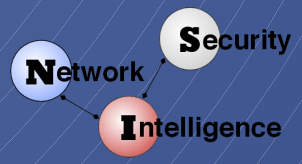
- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study



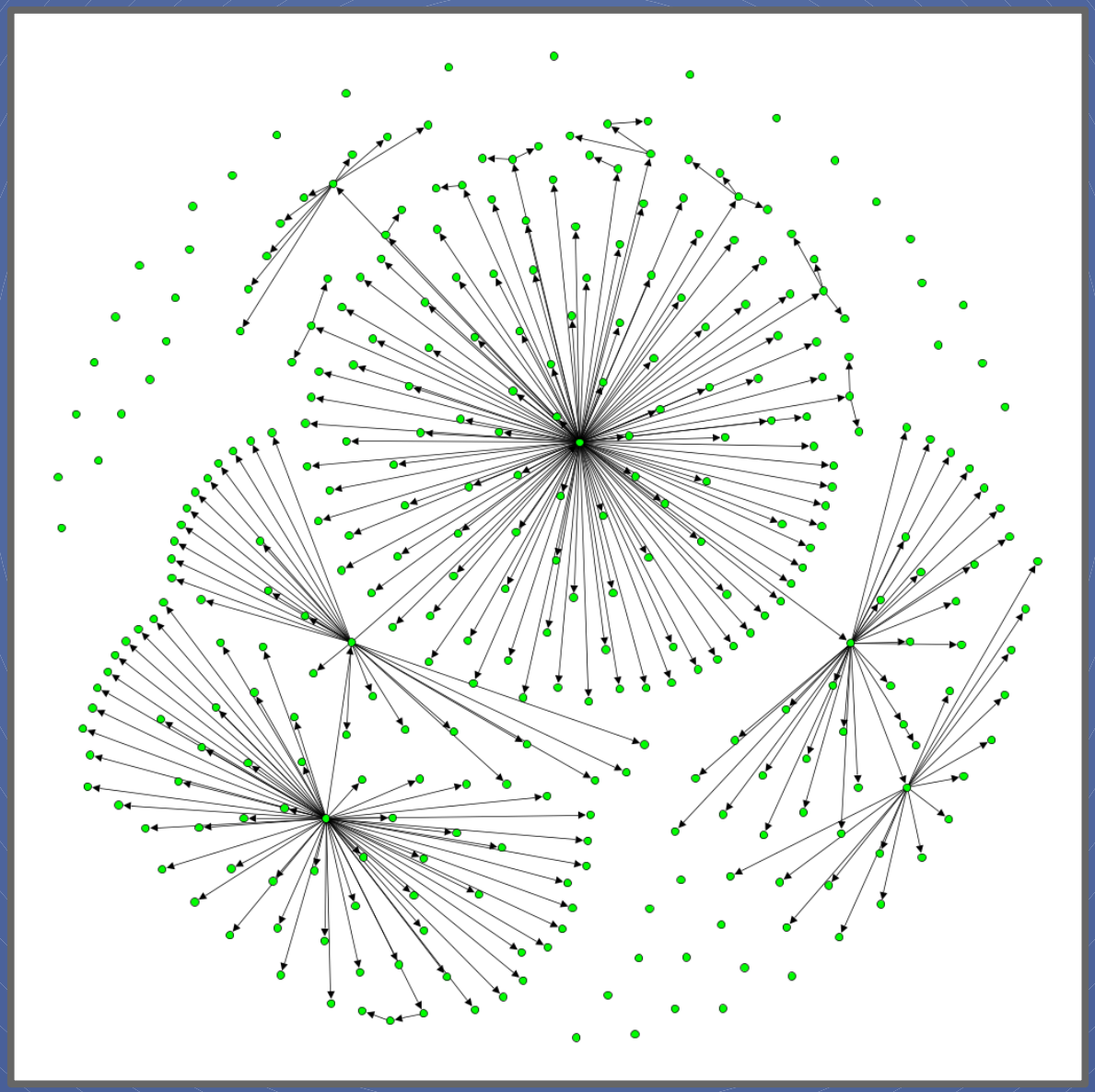
Case Study

Malware download incident

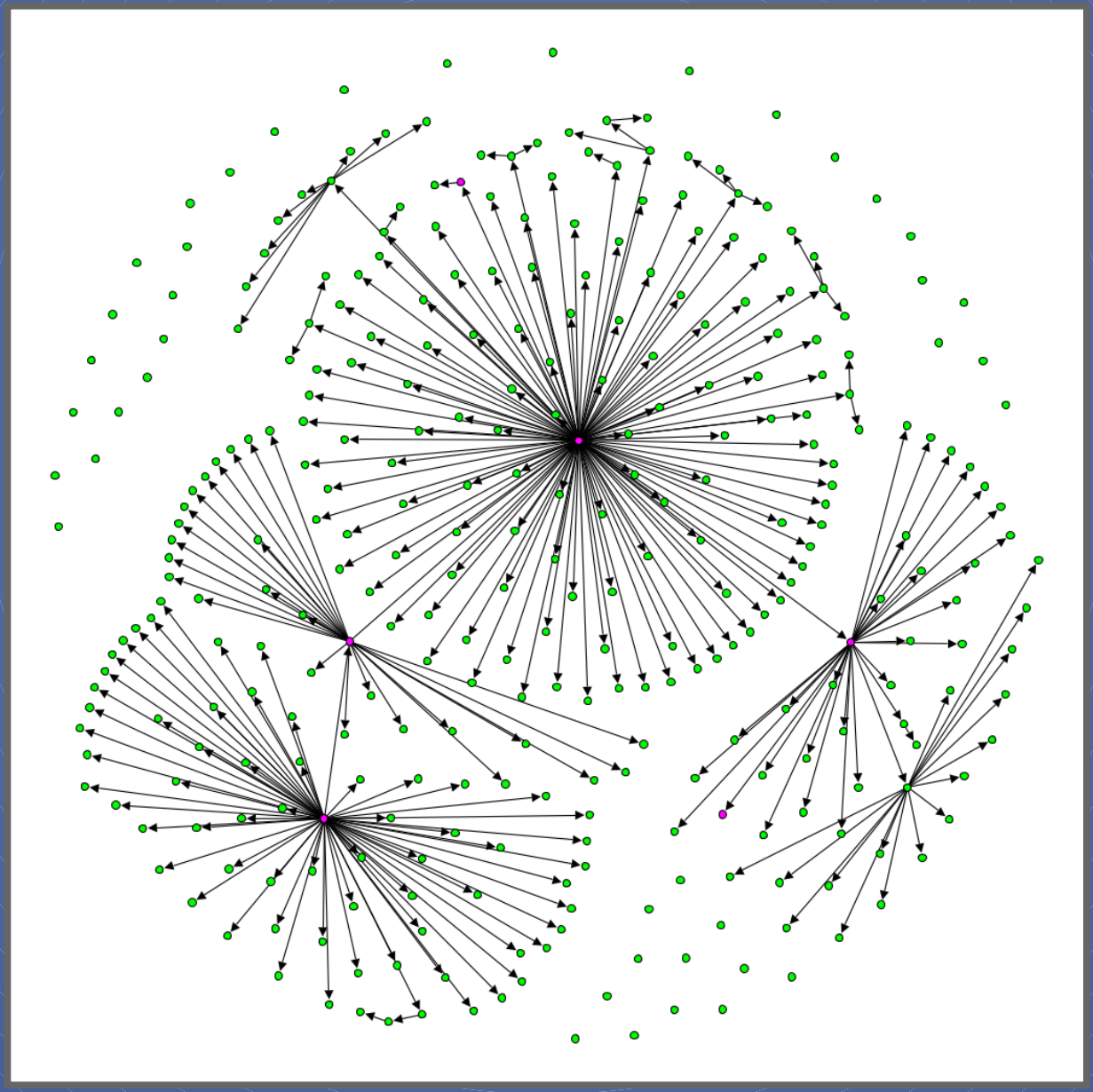
- Visited `bing.com`
- Searched with terms “far cry 3 hackz tools crack”
- Clicked on `allhackz[dot]net` from search results
- Clicked on “Download” button, opened two pages
  - `gameadvert[dot]com`
  - `wellmediaonline[dot]com`
- From `wellmediaonline[dot]com` download started via script from `effortlessdownload[dot]com`



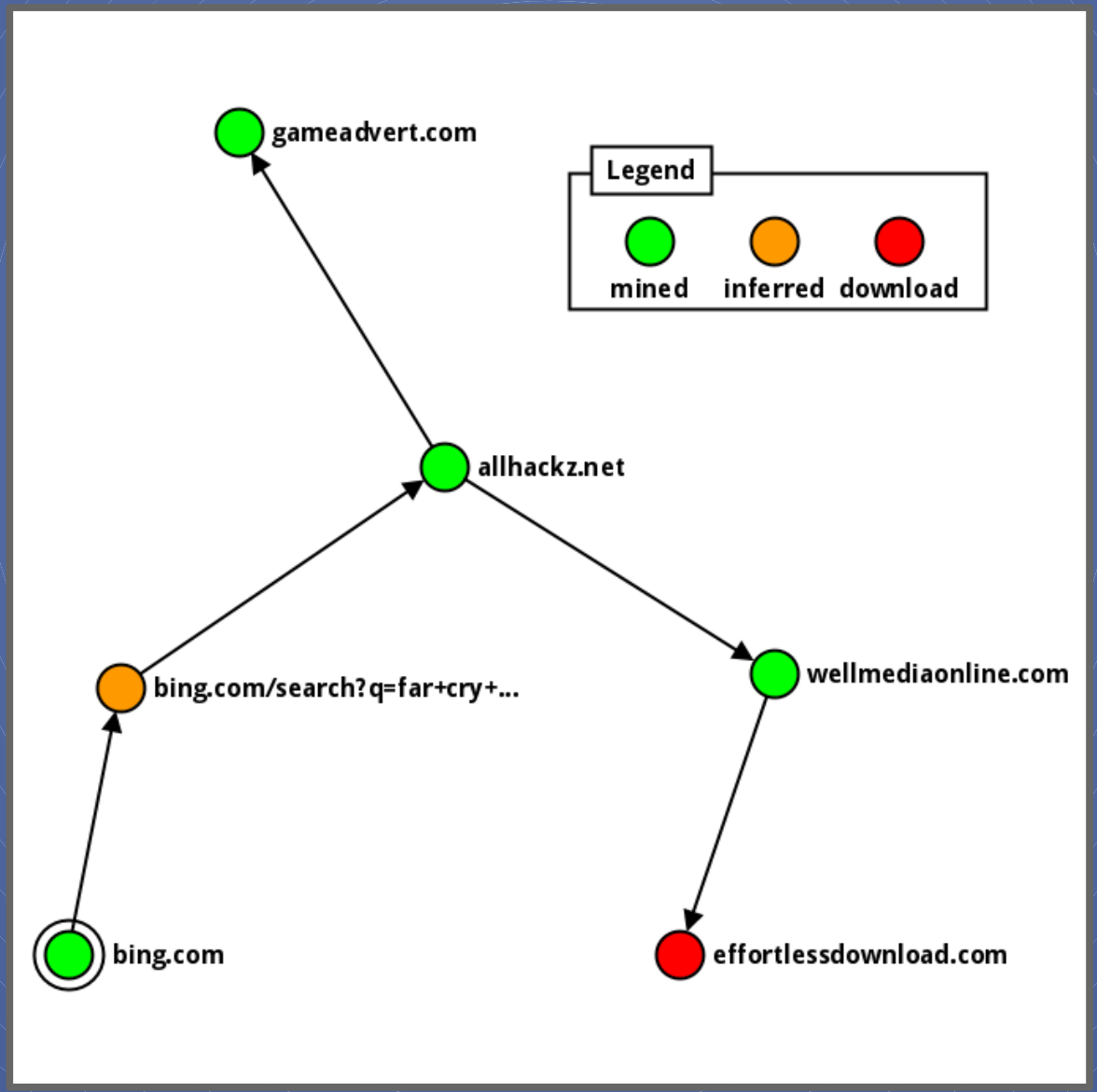
Case Study



Case Study



Case Study



References

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