Browser Fingerprinting from Coarse Traffic Summaries: Techniques and Implications

Ting-Fang Yen[1], Xin Huang[2],
Fabian Monrose[2], Michael K. Reiter[2]

1. Carnegie Mellon University, Pittsburgh, PA

2. University of North Carolina, Chapel Hill, NC

Application Fingerprinting

- Active or passive
- Prior work: Determining the *type* of application
    - File transfers, peer-to-peer, chat, etc.
      [Sen et al.'04; Karagiannis et al.'05; Hernandez-Campos et al.'05; Bernaille et al.'06]
    - Packet traces
    - Flow records
- Our work: Determining *specific implementations* of an application

Network Traffic Logging

- Monitoring network usage, traffic analysis, network intrusion detection...
- Flow records: Traffic summaries
  - Require less resources than recording packets
  - Uni- or Bi-directional
  - IP address, port numbers, protocol, timestamp, byte/packet counts

Browser Fingerprinting

- Our approach does not rely on payload
- Uses behavioral features evidenced in flows
- Implications: Improvements to …
  - Network intrusion detection systems
    - Platform-dependent malware
  - Traffic deanonymization
    - Identifying web **sites** in anonymized traffic

Challenges

- Browser traffic dependent on website content
  - Differences due to geographical locations
  - Differences over time
- Variations in user behavior …
  - Client browser configuration
  - Client hardware configuration
- How can we address these challenges?

PlanetLab Datasets

- Collected from 21 hosts across eight locations
  - Retrieve front page of top 150 websites over one month
  - Browser cache set to 400MB
- PlanetLab-Native Dataset
  - Firefox, Opera
- PlanetLab-QEMU Dataset
  - IE, Firefox, Opera, Safari

CMU Dataset

- Traffic from edge routers of Carnegie Mellon University campus network
- Six weeks from Oct-Dec 2007
- Argus flow records
    - Include first 64 bytes of flow payload
- Opera and Firefox
- Website retrievals identified by “GET / “, and include flows in the following 10 sec

Feature Selection

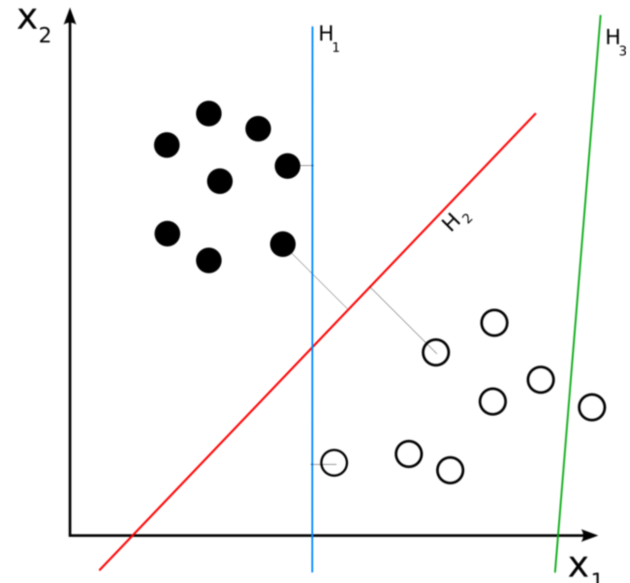
Flow Statistics	Byte count (in each direction) Packet count (in each direction) Flow duration Number of flows active simultaneously to this one Start time minus most closely preceding flow start time
Retrieval Statistics	Total number of flows Cumulative byte count from destination Cumulative flow duration Retrieval duration

- Mean, std.dev., max, min, median, first and third quartile, inter-quartile range, sum

- Feature selection using **information gain**
- Each retrieval represented by feature vector

Browser Classifier

- Support Vector Machine (SVM)
- Finds a hyperplane that maximally separates the data



- "Confidence":
  - Minimum distance of the testing instance to the hyperplane

Browser Classifier

- Train and test classifier on different datasets
- For each host *h*, returns the browser most classified in *h*'s retrievals

$$\text{Precision} = \Pr[\text{browser}(h) = b \mid \text{browserguess}(h) = b \neq \perp]$$

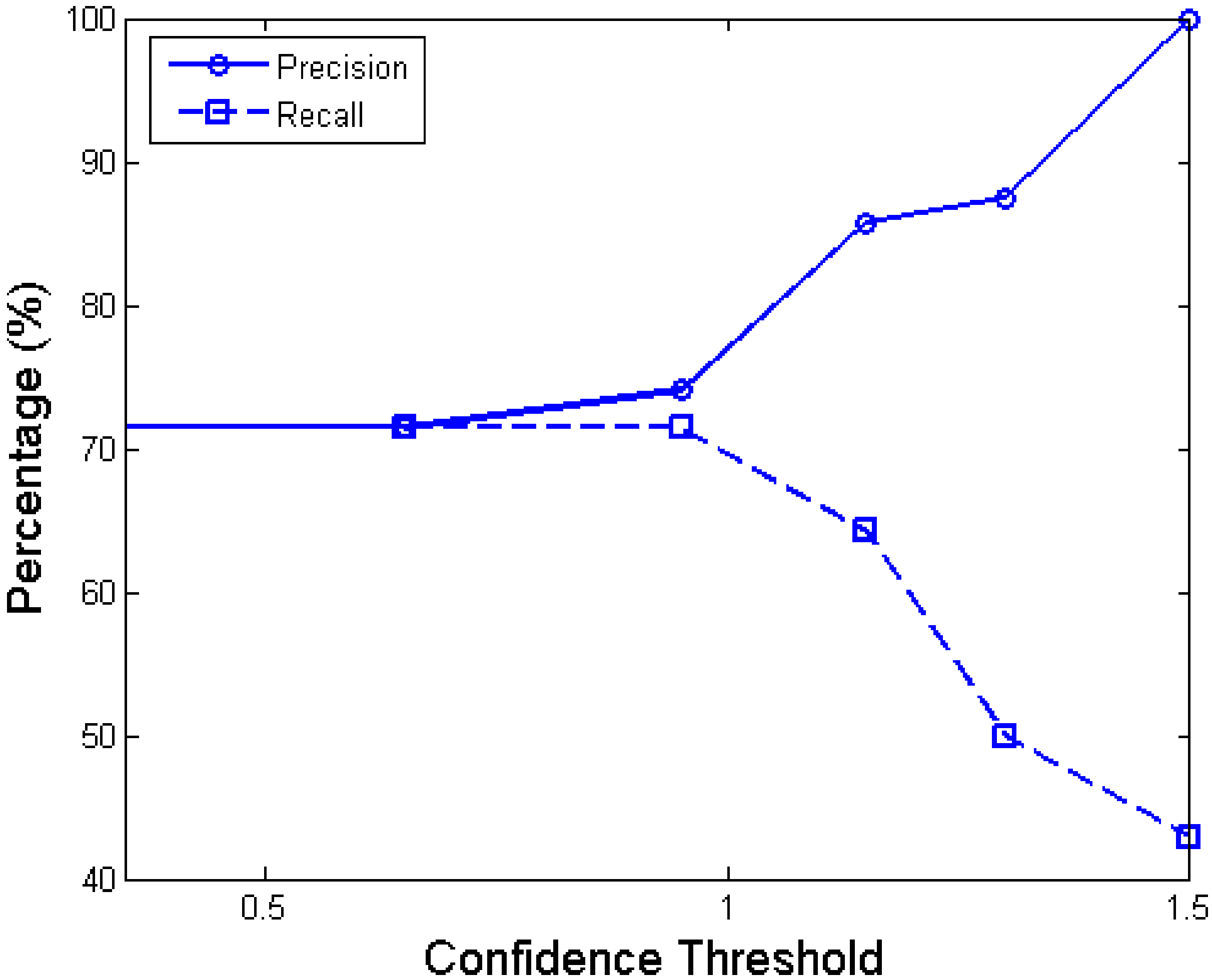
$$\text{Recall} = \Pr[\text{browserguess}(h) = b \mid \text{browser}(h) = b \neq \perp]$$

- browserguess(h) = $\perp$
  - Classifier makes no classification for host h
- browser(h) = $\perp$
  - Actual browser could not be determined

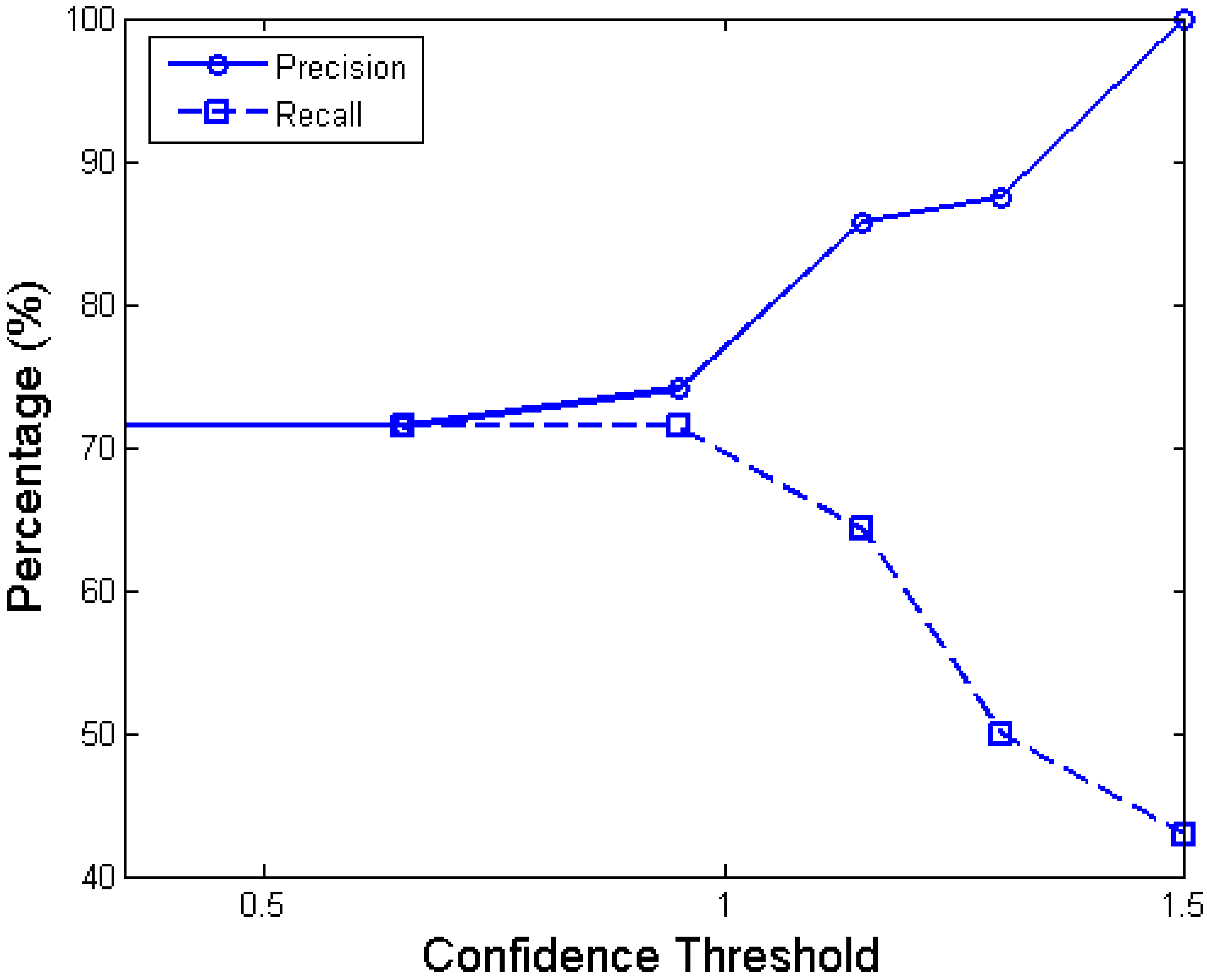
Tests on PlanetLab-QEMU

- Clean data in controlled environment
- Separate traffic by browser and location
- Training data
  - Traffic from top 100 websites
  - Traffic from all PlanetLab locations
- Testing data
  - Traffic from top 100-150 websites
  - Traffic from each PlanetLab location

Tests on PlanetLab-QEMU



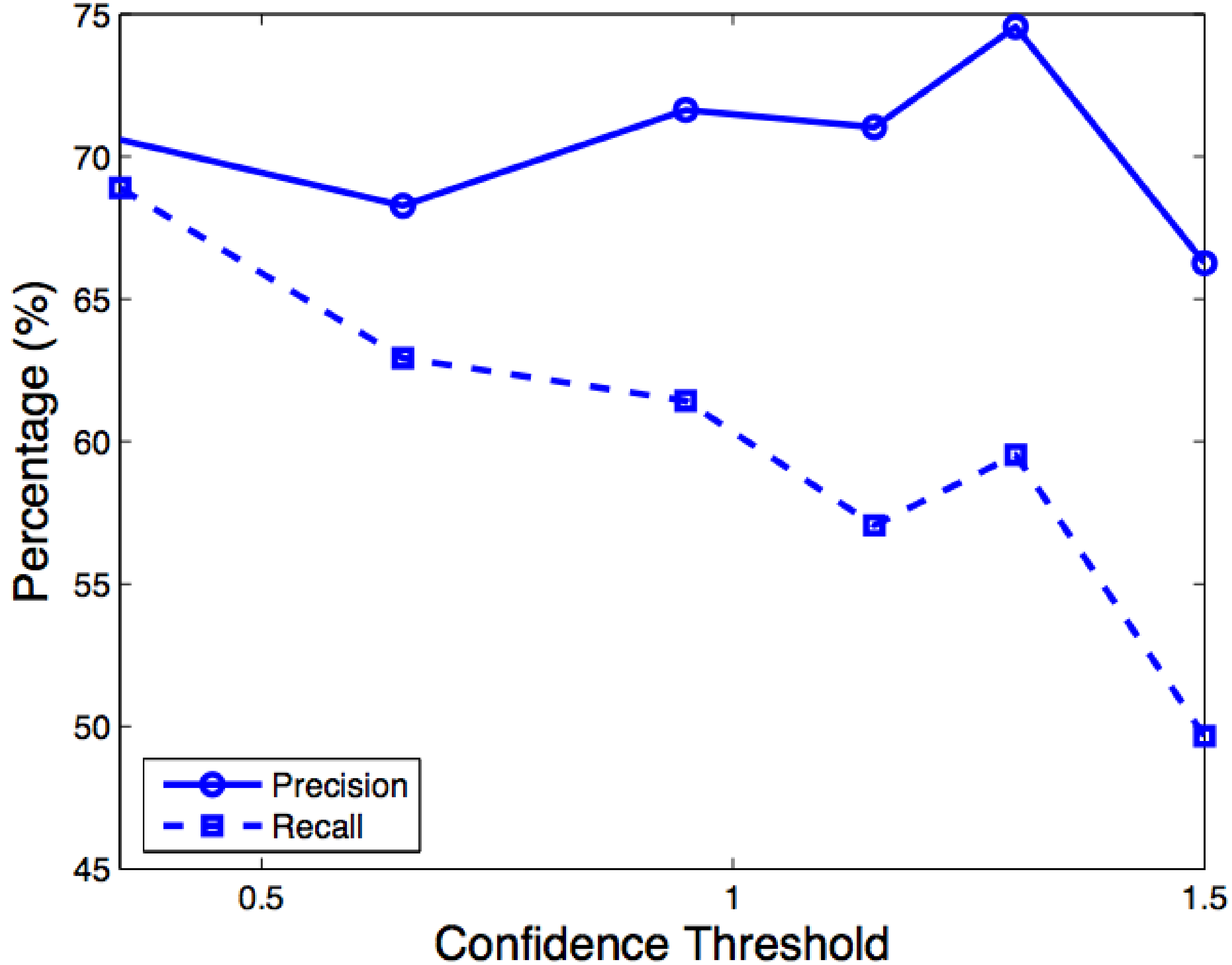
Tests on PlanetLab-QEMU



Pretty good, right? How about on real user traffic?

Tests on CMU Dataset

- Training data: PlanetLab-Native dataset

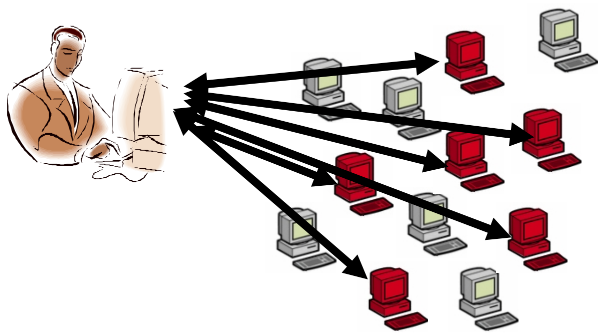


Browser Fingerprinting Works!

- Coarse traffic summaries
- Training and testing data from different geographical locations, different websites, different time frames
- Tests on real user data has **75%** precision and **60%** recall
    - Precision of random guessing is 25%

Applications to Network Intrusion Detection Systems

- Traffic Aggregation for Malware Detection (TAMD) [Yen and Reiter, DIMVA'08]
- Stealthy malware: spyware, adware, bots, …
    - Subtle command/control system
    - Organized malicious activities
        - Spamming, hosting phishing sites, DDoS attacks



Traffic Aggregation for Malware Detection

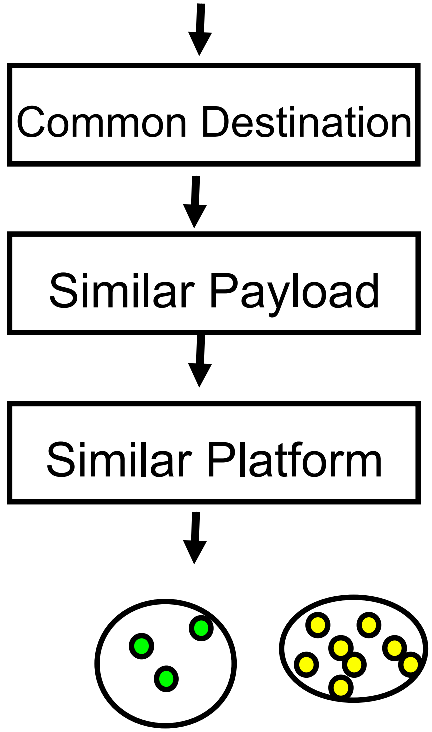
- Observe traffic at network border
  - Multiple infected hosts in the network
  - Malware communication patterns different from benign hosts
- Find traffic from multiple hosts that share similar characteristics
  - Common destination
  - Similar payload
  - **Similar platform**

Similar Platform

- Operating system specific features
  - Time-to-live (TTL) field, communications to characteristic sites (e.g., Microsoft time server)
- May fail to identify application-dependent malware
- Incorporate browser fingerprinting
  - Traffic sharing same OS *or* same browser

Evaluation

- Target platform-dependent infections that contact common destinations
- Output groups of traffic sharing multiple characteristics
- Data reduction tool

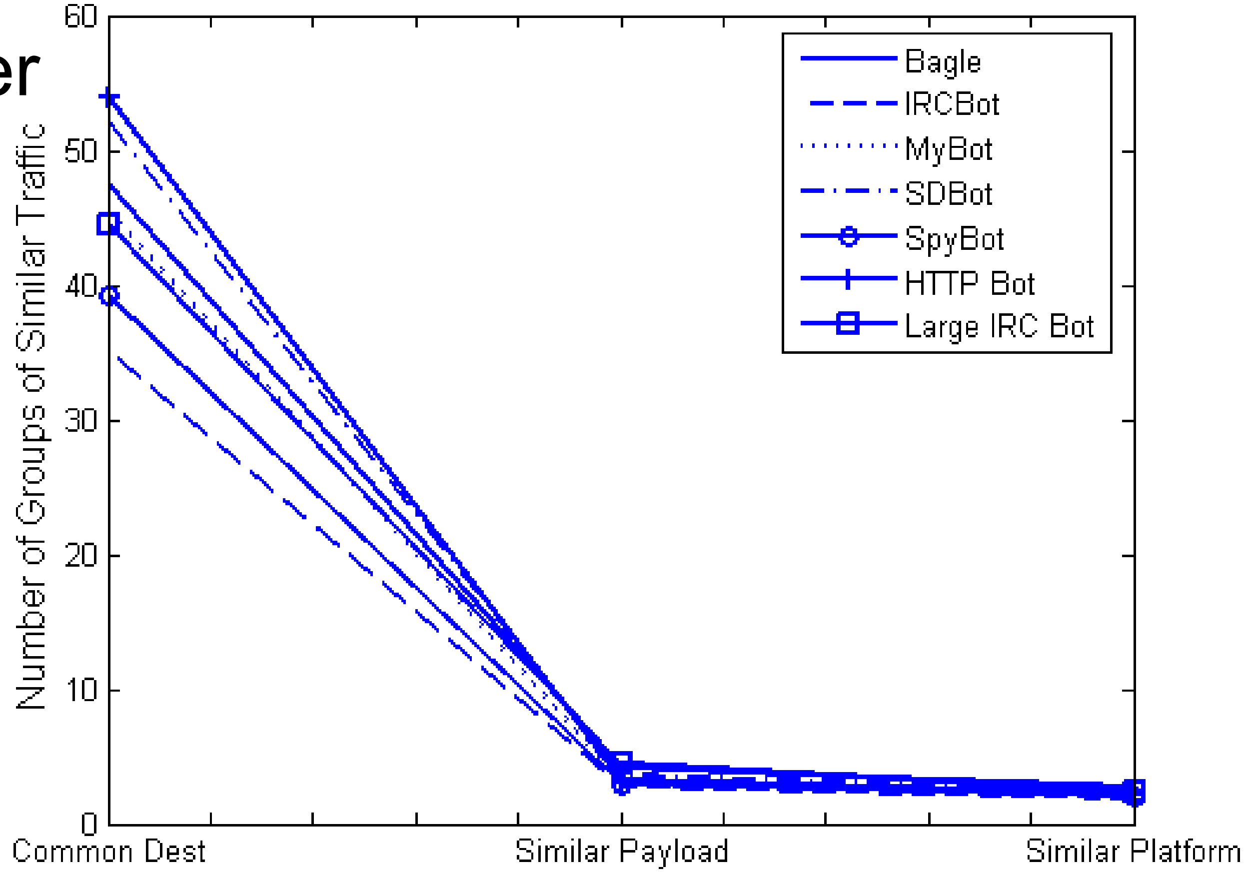


Evaluation

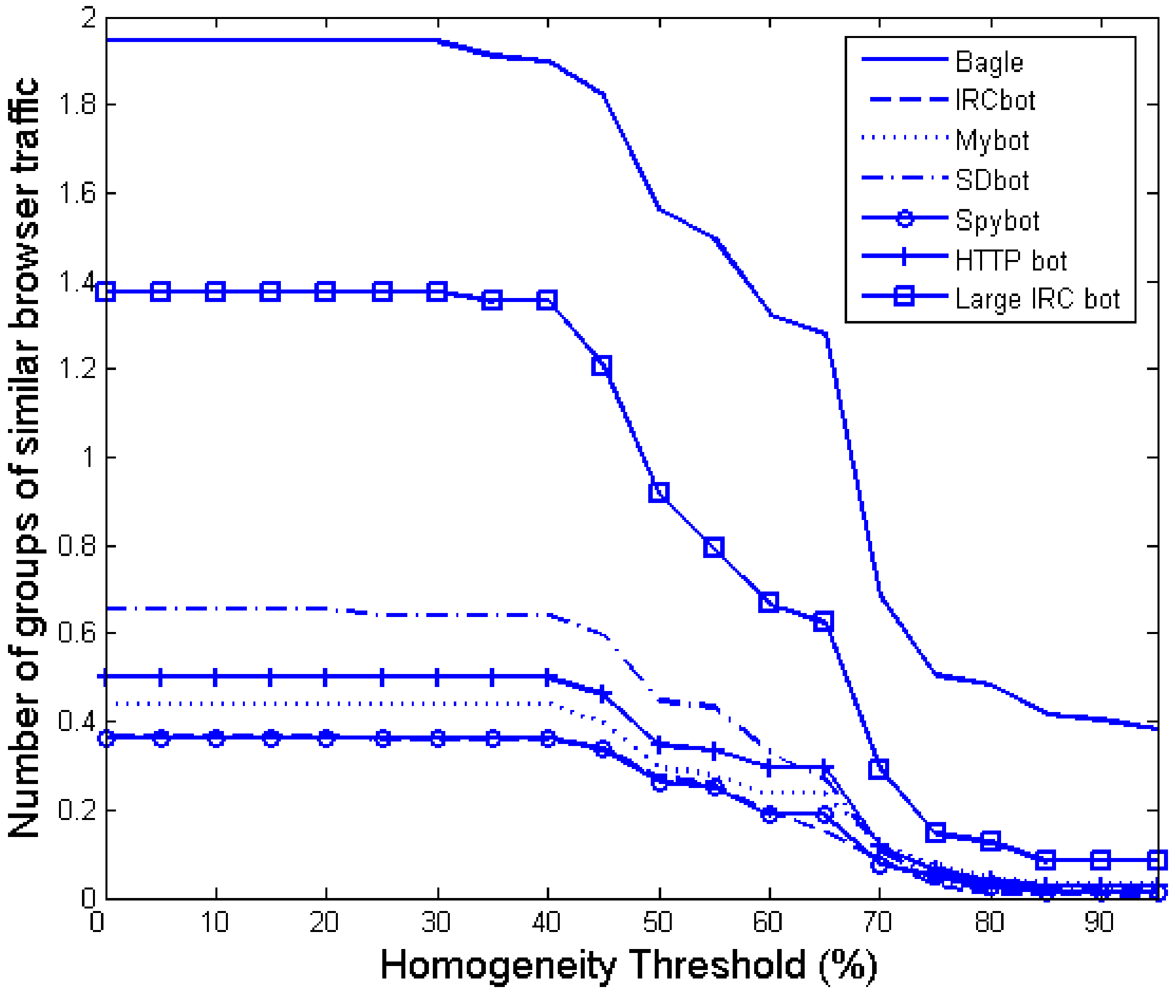
- Malware traffic:
  - Bagle, IRCBot, MyBot, SDBot, SpyBot, HTTP-based bot, large IRC botnet
- For every hour of traffic in CMU dataset
  - Assign malware traffic to originate from randomly selected internal hosts
  - Input to TAMD
  - Repeat for every hour, for each malware
- Malware are OS-dependent
  - Quantify *cost* of incorporating browser fingerprinting

Evaluation

- The hosts we assigned malware traffic to is always identified
- On average, 2.25 groups per hour



Evaluation



- 0.02 groups per hour due to browser similarity

Applications to Traffic Deanonymization

- Infers the web *sites* contacted in anonymized traffic
- Classifying browser first can improve precision of traffic deanonymization ...

Website Classifier

- Bayesian belief networks
  - Given a test instance, generates a probability for each class
  - Outputs class with highest probability
- Establishing “confidence”...
  - Only selects from probabilities above the “cutoff”

Website Classification Features

Flow Statistics	Byte count (in each direction) Packet count (in each direction) Flow duration Number of flows active simultaneously to this one Start time minus most closely preceding flow start time
Retrieval Statistics	Total number of flows Cumulative byte count from destination Cumulative flow duration Retrieval duration

- Per distinct server, for first five servers

Selecting Stable Websites

- Focus on stable websites
  - Determined by average number of flows and std. dev of byte/packet counts
  - Simple or high-variability websites do not include enough information for classifier to make confidence guesses
- 52 websites selected from top 100

Per-browser vs. Generic Classifier

- Per-browser website classifier
  - Trained on traffic from a single browser
- Generic website classifier
  - Trained on traffic from all four browsers
- Apply same testing data to compare results

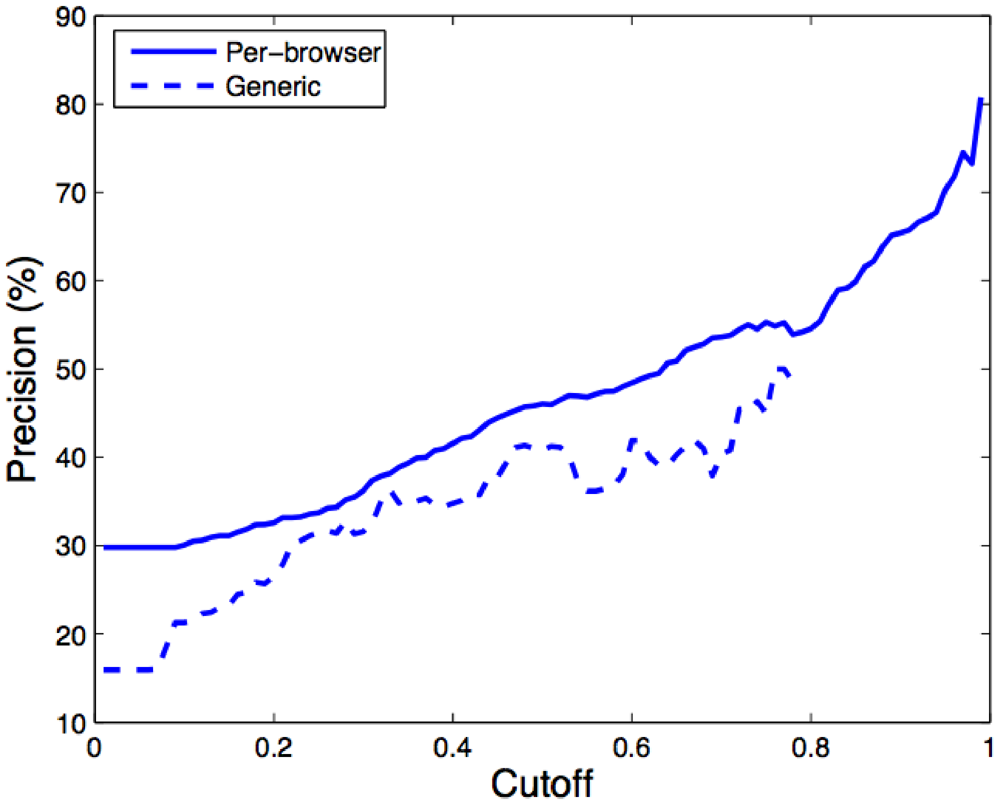
$$\text{Precision} = \Pr[\text{website}(r) = s \mid \text{websiteguess}(r) = s \neq \perp]$$

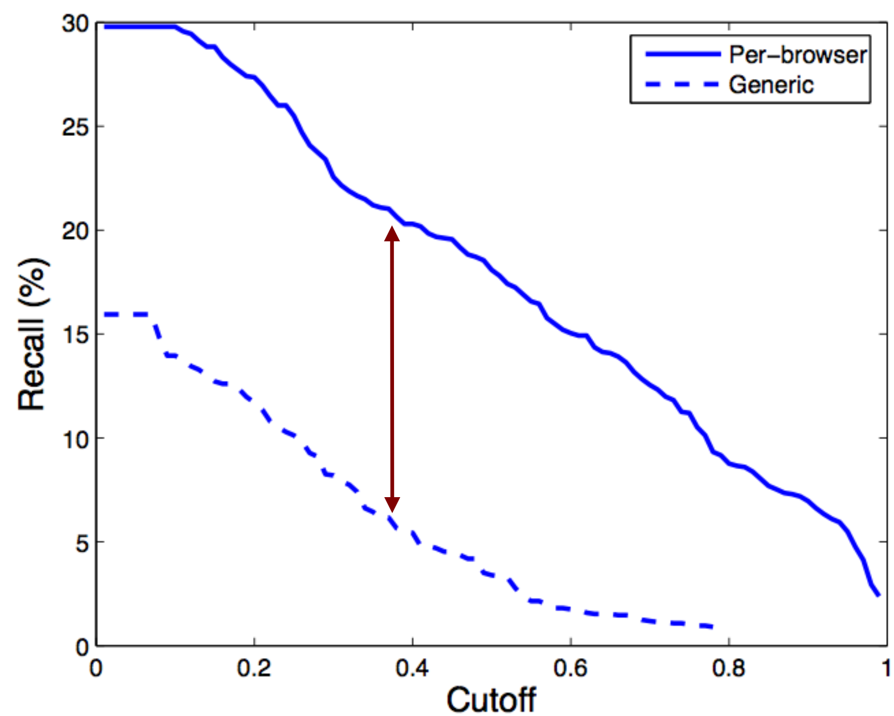
$$\text{Recall} = \Pr[\text{websiteguess}(r) = s \mid \text{website}(r) = s \neq \perp]$$

Tests on PlanetLab-QEMU

- Training data: Website retrievals from all PlanetLab locations
  - Per-browser website classifier for each browser
  - Generic website classifier
- Testing data: Website retrievals from each PlanetLab location
- Which per-browser website classifier?
  - Determined by browser fingerprinting

Tests on PlanetLab-QEMU

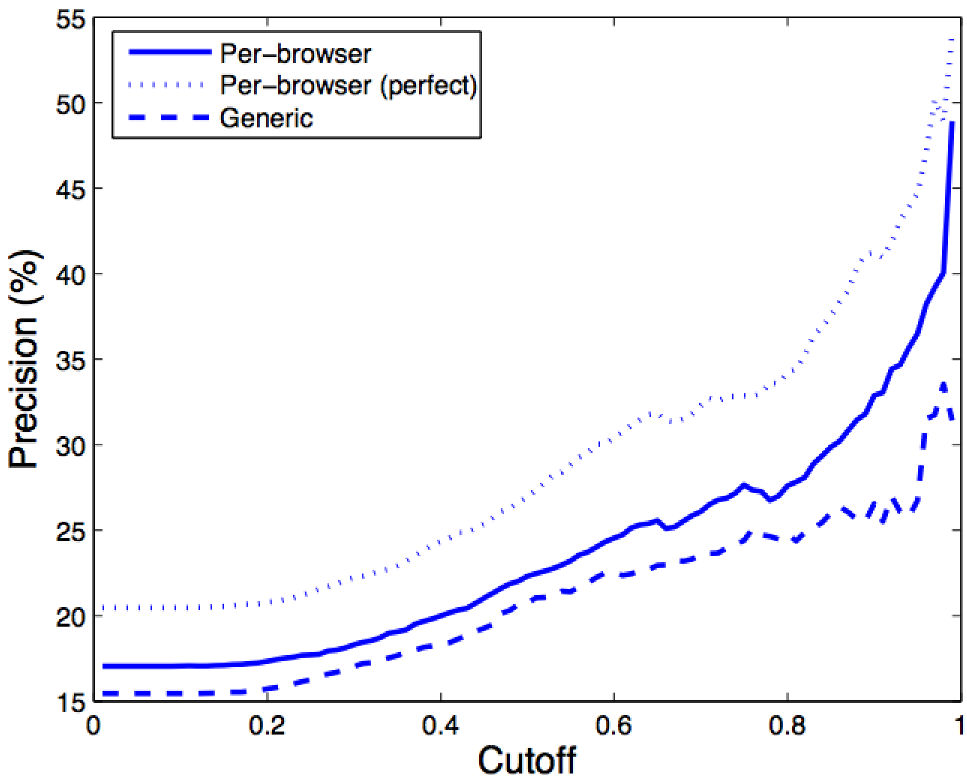


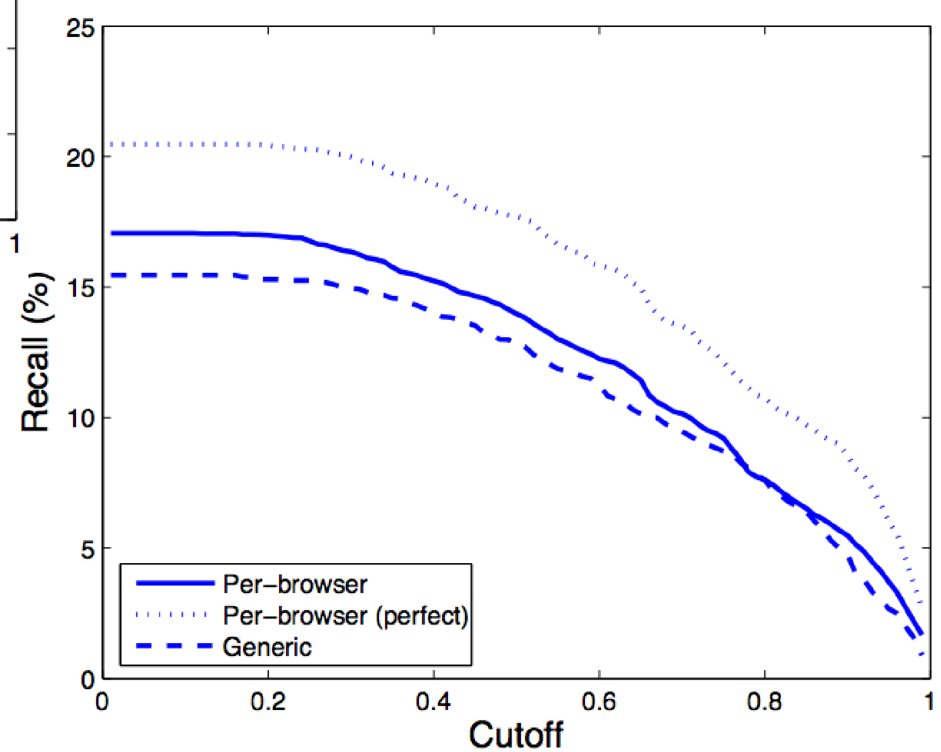


Tests on CMU Dataset

- Training data: PlanetLab-Native dataset
- Testing data: CMU dataset
  - Ground truth from HTTP “Host” field
- Which per-browser website classifier?
  - Determined by browser fingerprinting
  - Actual browser implementation
    - Show improvements when more accurate browser fingerprinting can be developed

Tests on CMU Dataset





Implications for Traffic Deanonymization

- When focusing on specific websites of interest to the attacker...

Website	Precision (%)		Recall (%)	
	Per-browser	Generic	Per-browser	Generic
`adobe.com`	17.59	0.00	9.55	0.00
`dailymotion.com`	84.62	57.05	50.00	44.95
`nytimes.com`	21.15	16.26	12.26	9.13
`wordpress.com`	13.98	0.00	7.15	0.00
`yahoo.com`	45.52	29.60	29.81	19.78

Conclusion

- Browser fingerprinting on flow records reached 75% precision and 60% recall
- Enables network intrusion detection system to detect more malware
- Improves precision of traffic deanonymization