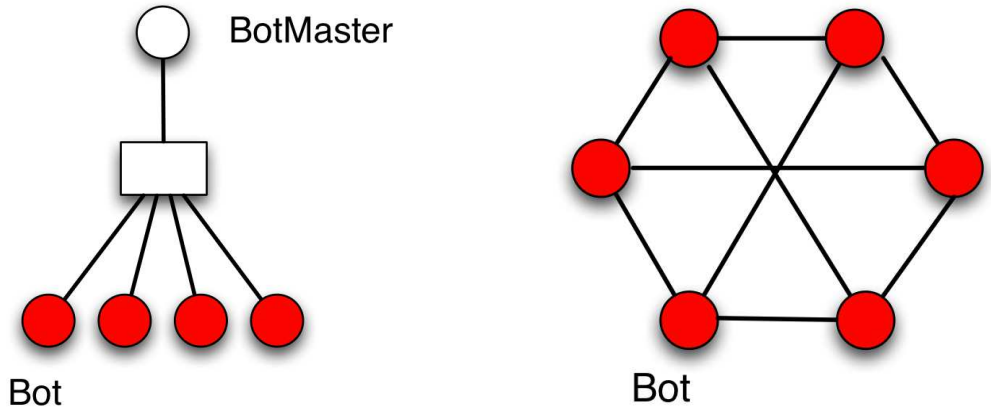
Outline

- Motivation
- Our System
- Evaluation
- Conclusion

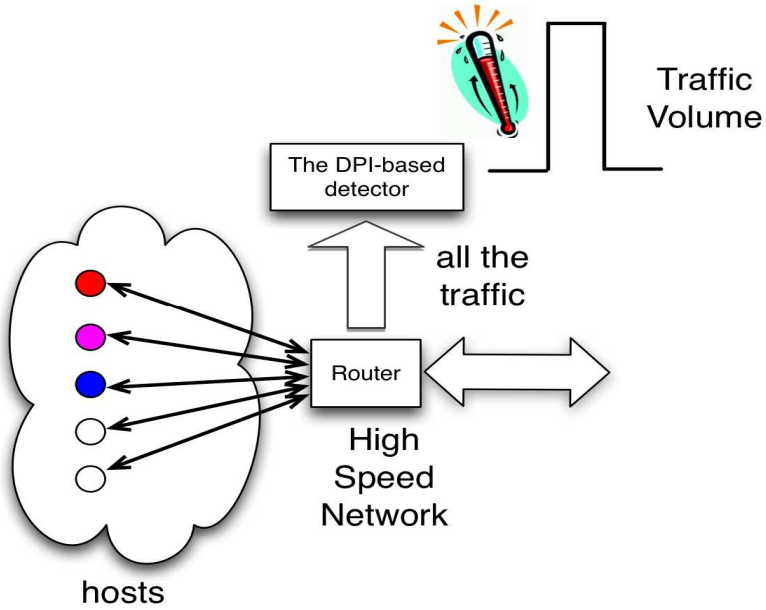
Botnet

- A botnet is a collection of bots controlled by a botmaster via a command and control (C&C) channel
  - Centralized C&C, P2P-based C&C
- Botnets serve as the infrastructures for a variety of attacks
  - Exploiting, scanning, spamming, phishing, DDoS, etc.
- Botnet detection is of great importance



Motivation

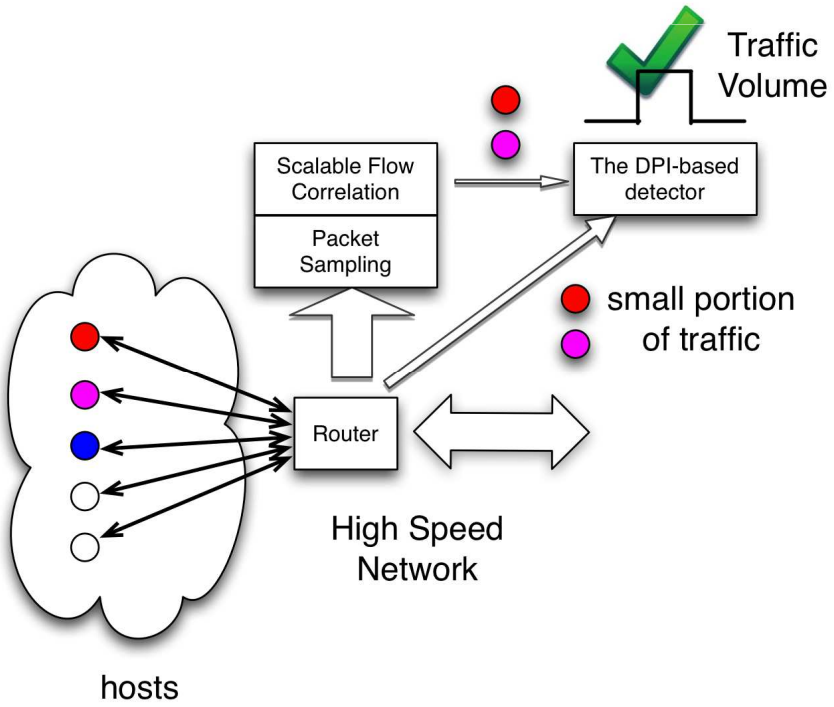
- Current detection approaches are based on Deep Packet Inspection (DPI)
  - BotHunter [Security 07]
  - BotSniffer [NDSS 07]
  - BotMiner [Security 08] (malicious activity plane)
  - TAMD [DIMVA 08]



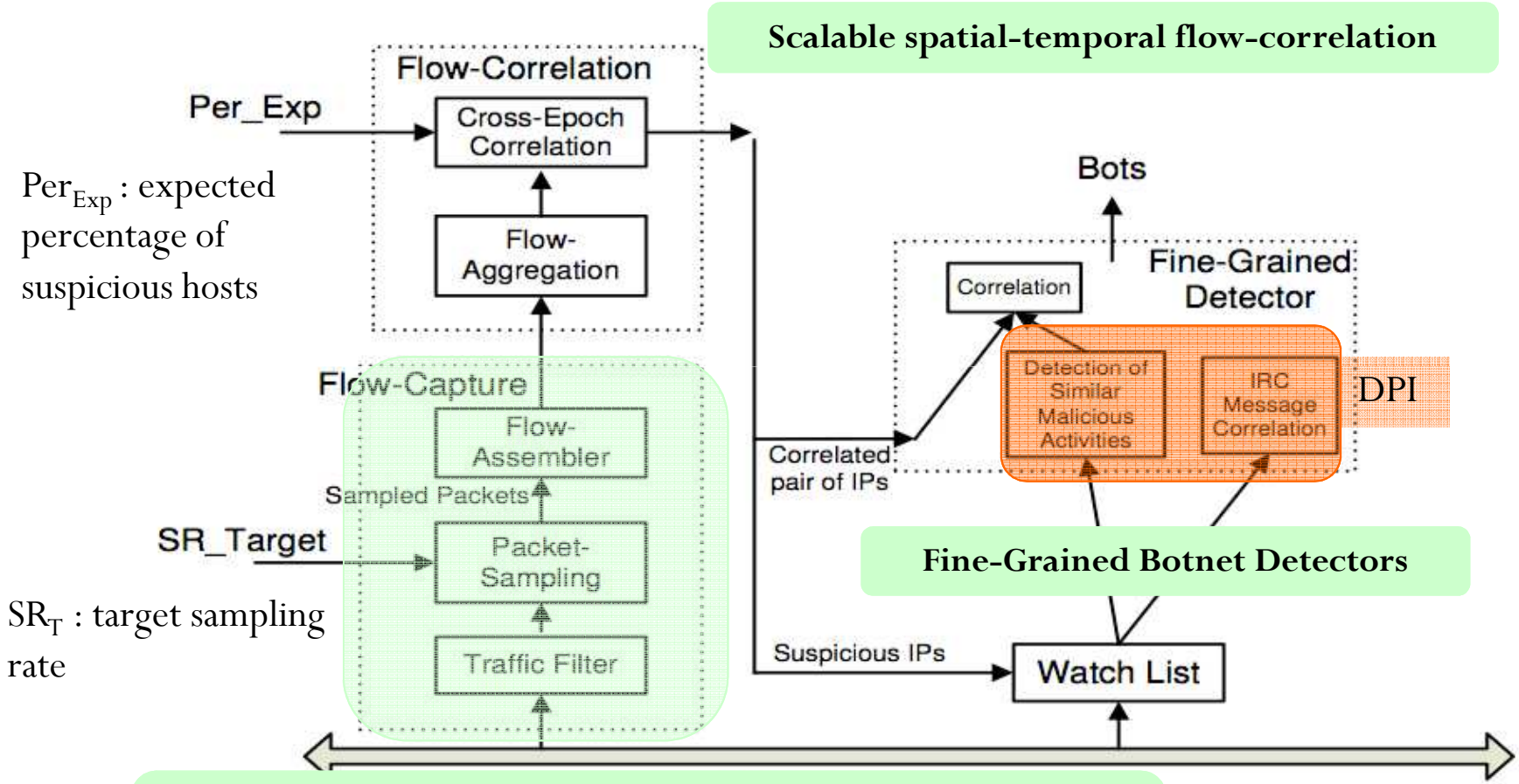
Not Scalable for high-speed and high-volume networks!

Our system

- A layered traffic analysis approach
  1. Identify suspicious hosts from high speed network through flow-correlation
     - Botnet-aware packet sampling algorithm (B-Sampling)
     - Scalable spatial-temporal flow-correlation algorithm
  2. Apply Fine-grained DPI-based detectors to suspicious hosts



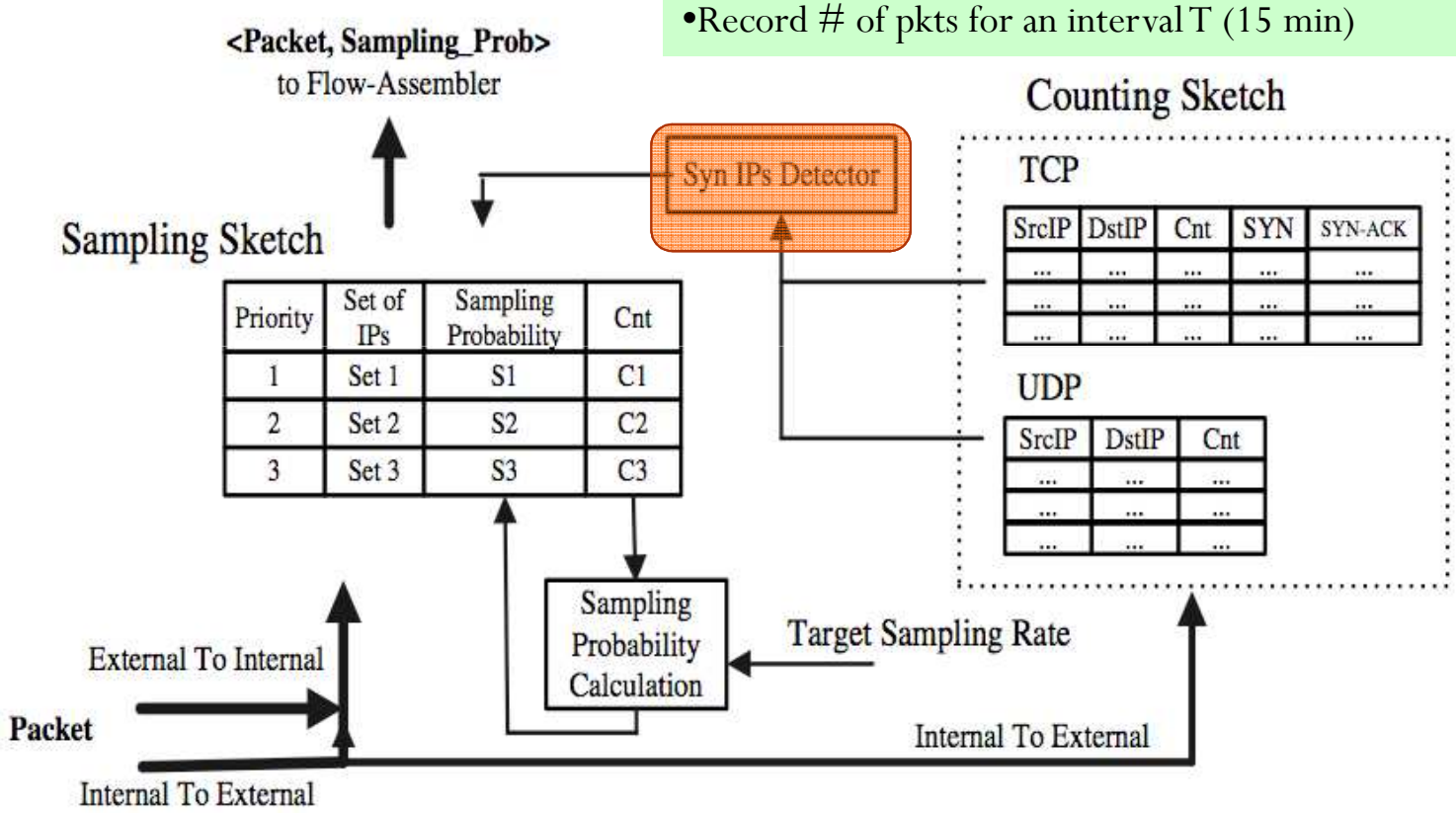
System Architecture



•Flow Capture:
1.B-Sampling: Botnet-Aware Adaptive Packet Sampling
2.Flow-Assembler

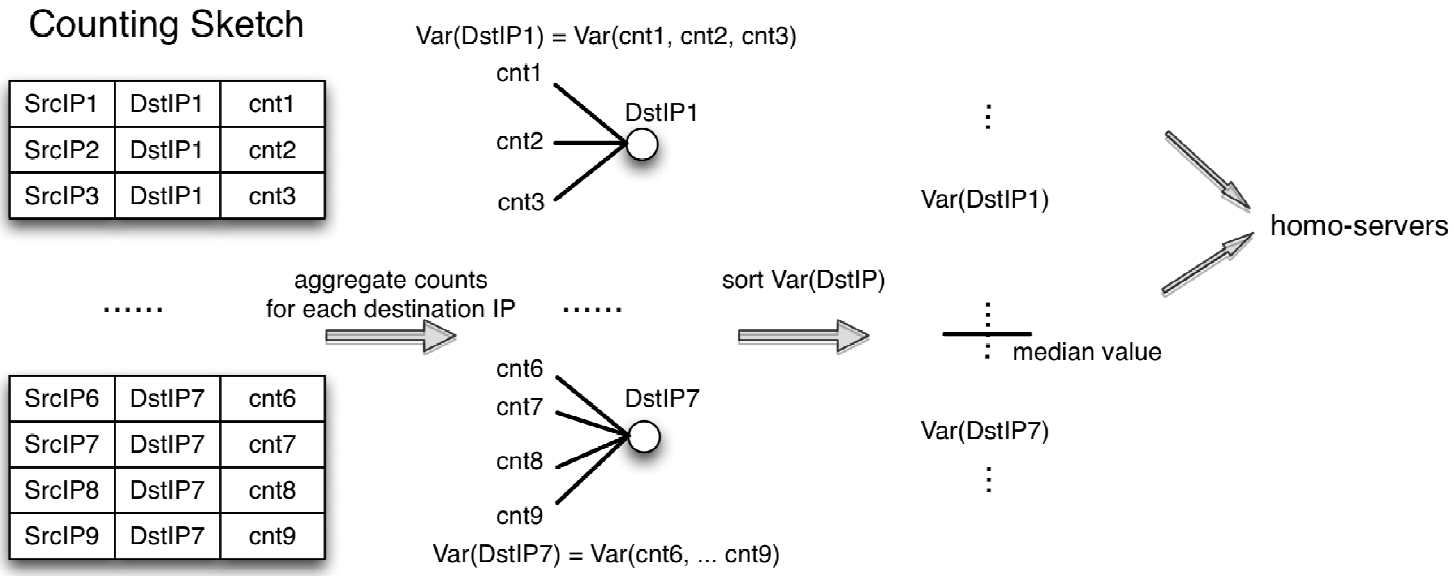
Flow Capture

- Indexed by Hash(SrcIP||DstIP)
- Record # of pkts for an interval T (15 min)



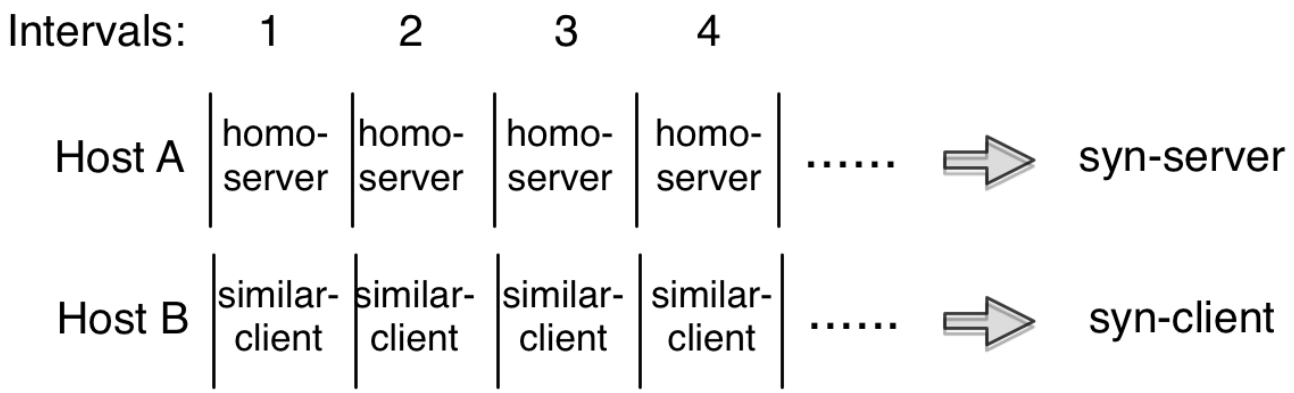
Flow Capture: Synchronized IPs Detector

- homo-servers
  - Hosts outside the monitored networks whose clients show small variance of connections in a time interval (T=15 min)
- similar-clients
  - Hosts within the monitored networks that generate similar connections to a large number of destination IPs in a time interval (T=15 min)



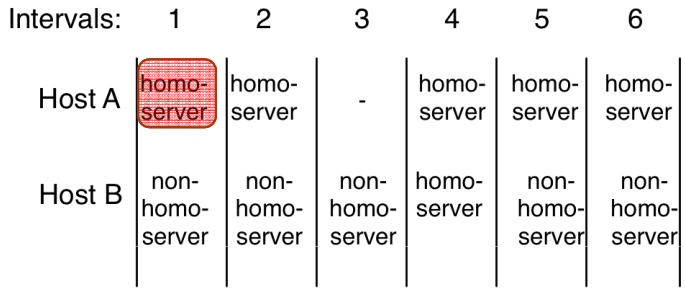
Flow Capture: Synchronized IPs Detector

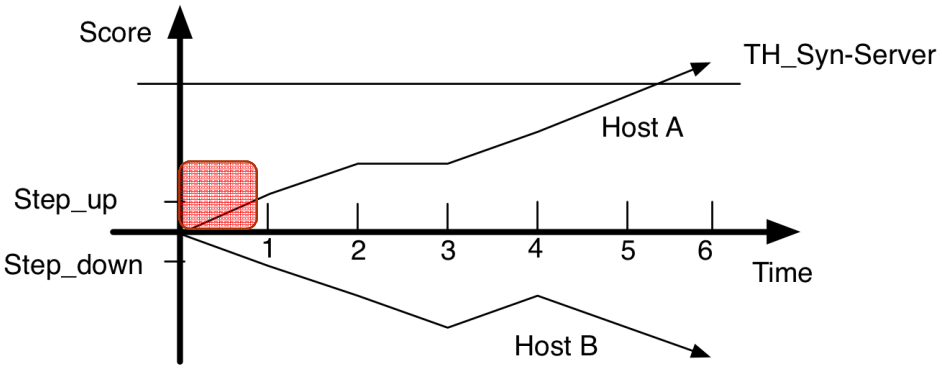
- From homo-servers and similar-clients, we identify:
  - syn-servers
    - C&C servers for centralized-based botnets
  - syn-clients
    - Bots of P2P-based botnets



Flow Capture: Synchronized IPs Detector

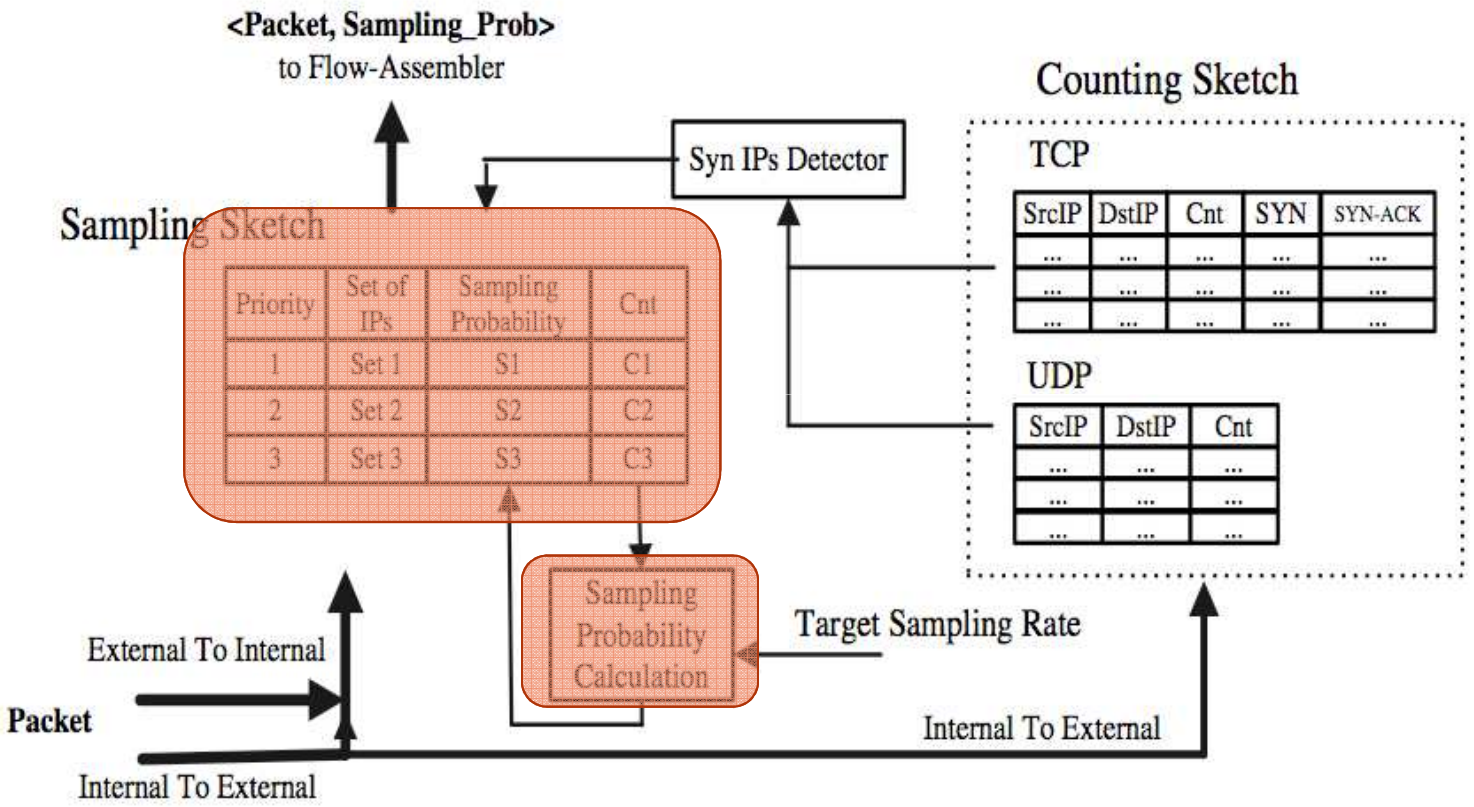
- Identify syn-server/client based on home-server/similar-client





$$step_{up} = 1$$
$$step_{down} = 0.2$$
$$TH_{syn-server} = 4$$

Flow Capture: Sampling Probability Calculation



Why we need a new sampling algorithm?

- Uniform sampling or periodic sampling
  - Prune to capturing packets in large flows and missing small flows (e.g. netflow)
- FlexSample [IMC 08]
  - Samples more packets from specific traffic subpopulations based on programmable conditions (e.g. small and medium flows)
  - The diversity of C&C communications of different botnets makes it challenging to set conditions for FlexSample to sample packets from a wide range of botnets.
- Requirement
  - Let the real sampling rate be close to target sampling rate.
  - Sample more packets from C&C communication flows.

Flow Capture: Sampling Probability Calculation

- The Priority-based Sampling Algorithm

Sampling Sketch

Priority	Set of IPs	Sampling Rate	Cnt
1	syn-servers	p1	c1
2	syn-clients	p2	c2
3	Others	p3	c3

Algorithm 2: Priority-based Sampling Algorithm

Input: $P_t, f_1, f_2, \ldots, f_n$
Output: $p_1, p_2, \ldots, p_n$
begin
  $budget = 1$;
  foreach $i = 1 \ldots n$ **do**
    if $f_i == 0$ *or* $budget \leq$ (
      $p_i = 0$;
      continue;
    else
      $p_i = budget * \frac{P_t}{f_i}$;
      $p_i = p_i > 1?1 : p_i$;
      $budget- = p_i * \frac{f_i}{P_t}$;
  return $\{p_1, p_2, \ldots, p_n\}$;
end

$$p_i = budget * \frac{P_t}{f_i}$$
$$p_i = p_i > 1?1 : p_i$$
$$budget_i = p_i * \frac{f_i}{P_t}$$
$$budget = budget - budget_i$$

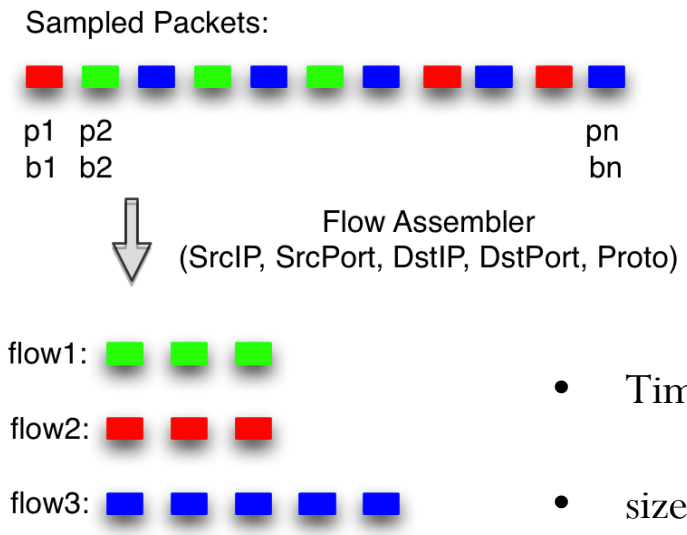
Estimate f_i^{curr}

$$SR_{Actual} = \sum_{i=1}^{n} f_i * p_i = \sum_{i=1}^{n} f_i * \frac{budget_i * P_t}{f_i} = P_t \sum_{i=1}^{n} budget_i = P_t$$

- P_t: the pre-defined target sampling rate
- f_i: the packet fraction for a priority out of all the packets $f_i = w_1 f_i^{prev} + w_2 f_i^{curr}$
- $budget_i$: the fraction of the sampled packets we would like to give to a particular priority
- $budget$: available budget

Flow Capture: Flow Assembler

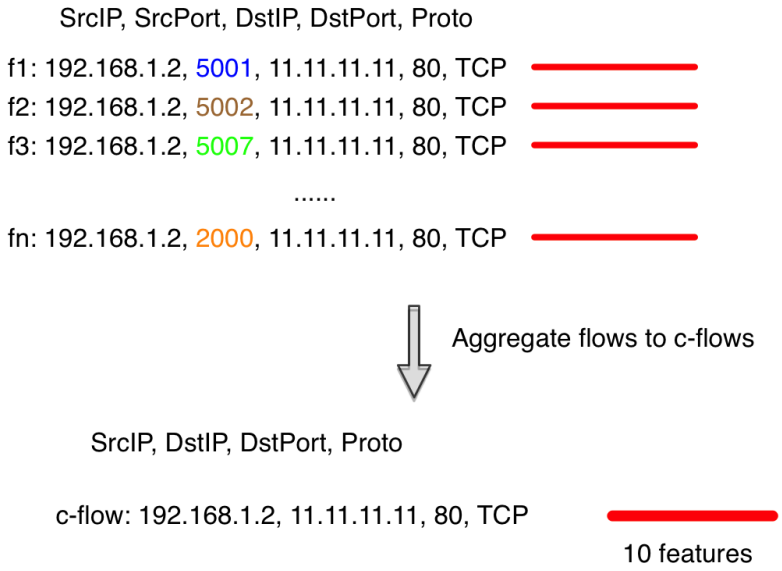
- Assemble each sample packet, together with its sampling rate (p_i), to 5-tuple flows identified by (SrcIP, SrcPort, DstIP, DstPort, Proto)



- $\text{Time}_{\text{start/end}}$: the start/end time of this flow
- $\text{size}_{\text{actual}}$: flow size, $size_{Actual} = n$
- $\text{byte}_{\text{actual}}$: the # of bytes observed, $byte_{Actual} = \sum_{i=1}^{n} b_i$
- size_{est}: the estimated flow size, $size_{Est} = \sum_{i=1}^{n} \frac{1}{p_i}$

Flow Correlation: Get C-flows

- C-flow
  - Aggregates a set of 5-tuple flows sharing the same tuple of (SrcIP, DstIP, DstPort, Proto) in a certain epoch (12 hours).
  - Represents the communication pattern from a host to a remote host and port in a certain epoch.

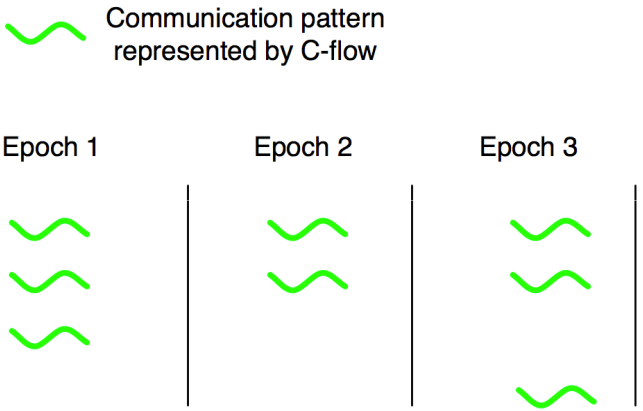


10 feature-vector to represent a C-flow

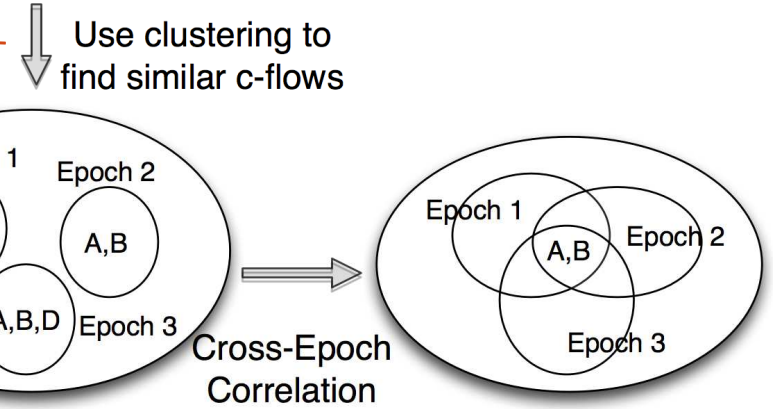
- the means and variances of
  - "# of flows per hour"
  - "# of packets per flow"
  - "# of packets per second"
  - "# of bytes per packet"
- fph_{max}: the maximum number of flows per hour
- $time_m$: the median time interval of two consecutive flows

Flow Correlation: Cross-Epoch Correlation

- If a pair of hosts share similar communication patterns for at least M out of N epochs (M<=N), they are suspicious.

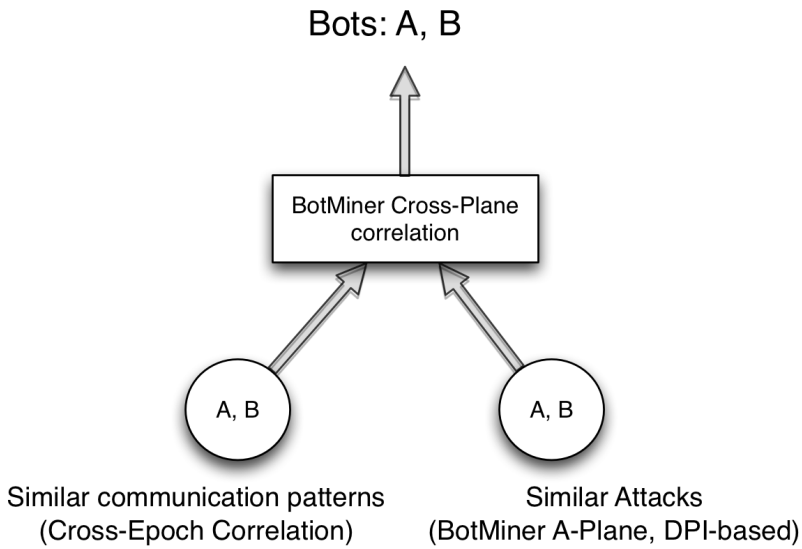


- Birch, a streaming clustering algorithm with good scalability
- Increase the "diameter" for discovering clusters to identify up to Per_{Exp} hosts



Fine-Grained Detectors

- Fine-Grained detectors only focus on traffic of Per_{Exp} hosts for deep packet inspection
  - If a pair of hosts share persistently similar communication patterns and commit similar attacks, they are identified as bots.(a modified version of BotMiner)
  - BotSniffer's IRC-based C&C detection component



Evaluation

- Experimental Data

Trace	# of Pkts	Dur	Info
Mar25	205,079,914	12h	header
Mar26	280,853,924	24h	header
Mar27	318,796,703	24h	header
Mar28	444,260,179	24h	header
Mar31	102,487,409	1.5h	full

Table 1: Background Traces

Trace	Dur	Bots
Bot-IRC-A	4days	3
Bot-IRC-B	4days	4
Bot-HTTP-A	4days	3
Bot-HTTP-B	4days	4
Bot-HTTP-C	4days	4
Bot-P2P-Storm	4days	2
Bot-P2P-Waledac	4days	3

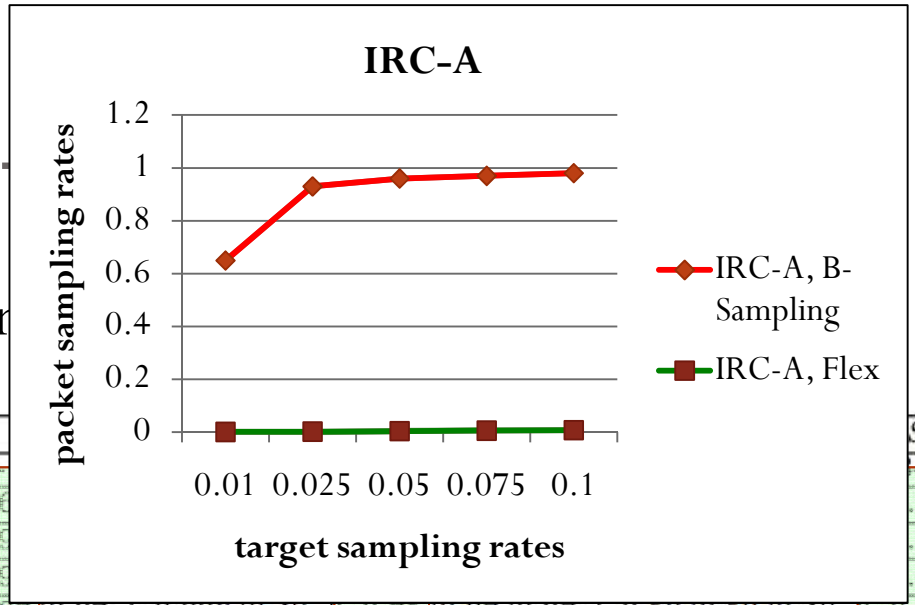
Table 2: Botnet traces

- Experimental Setup
  - 12 hours for each epoch; totally 7 epochs
  - If a pair of hosts share similar communication patterns 3 epochs out of 7 epochs, they are identified to share "persistently similar communication patterns".

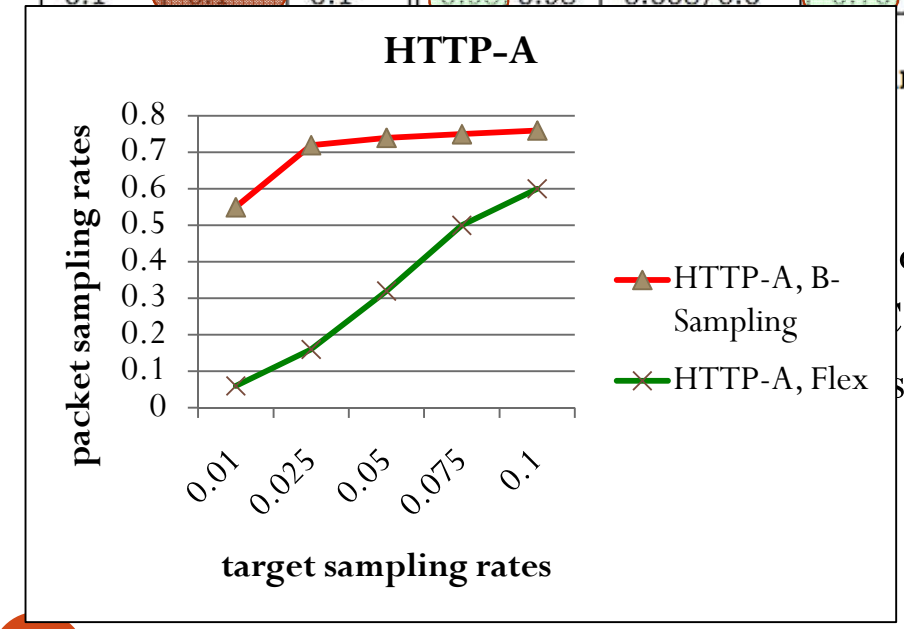
Both-IRC-A: TR/Agent.1199508.A
Bot-HTTP-A: Swizzor.gen.c
Bot-P2P-Storm: storm
Bot-P2P-Waledac: waledac
Others from RuBot

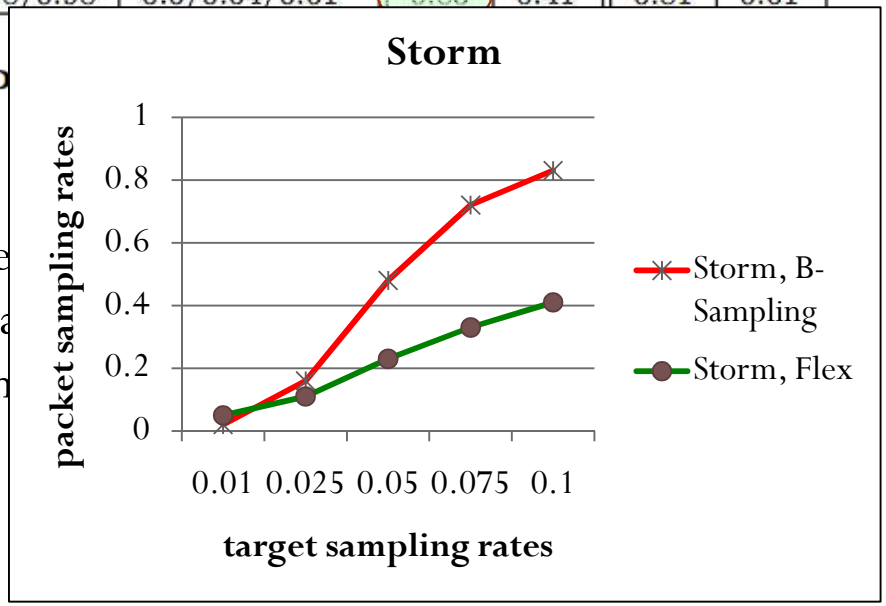
Evalua

- Packet Sar



SR_T	SR_{Actual}					SR_{Storm}		$SR_{Waledac}$	
	B-	Flex	B-				Flex	B-	Flex
0.01	0.012	0.01	0.6			.02	0.05	0.02	0.07
0.025	0.027	0.025	0.9			.16	0.11	0.18	0.16
0.05	0.052	0.05	0.9			.48	0.23	0.48	0.33
0.075	0.076	0.075	0.97/0.97	0.006/0.48	0.75/0.97/0.97	0.50/0.50/0.48	0.72	0.33	0.7	0.48
0.1	0.1	0.1	0.98/0.98	0.008/0.6	0.76/0.98/0.98	0.6/0.64/0.61	0.83	0.41	0.81	0.61





Evaluation

- Cross-Epoch Correlation

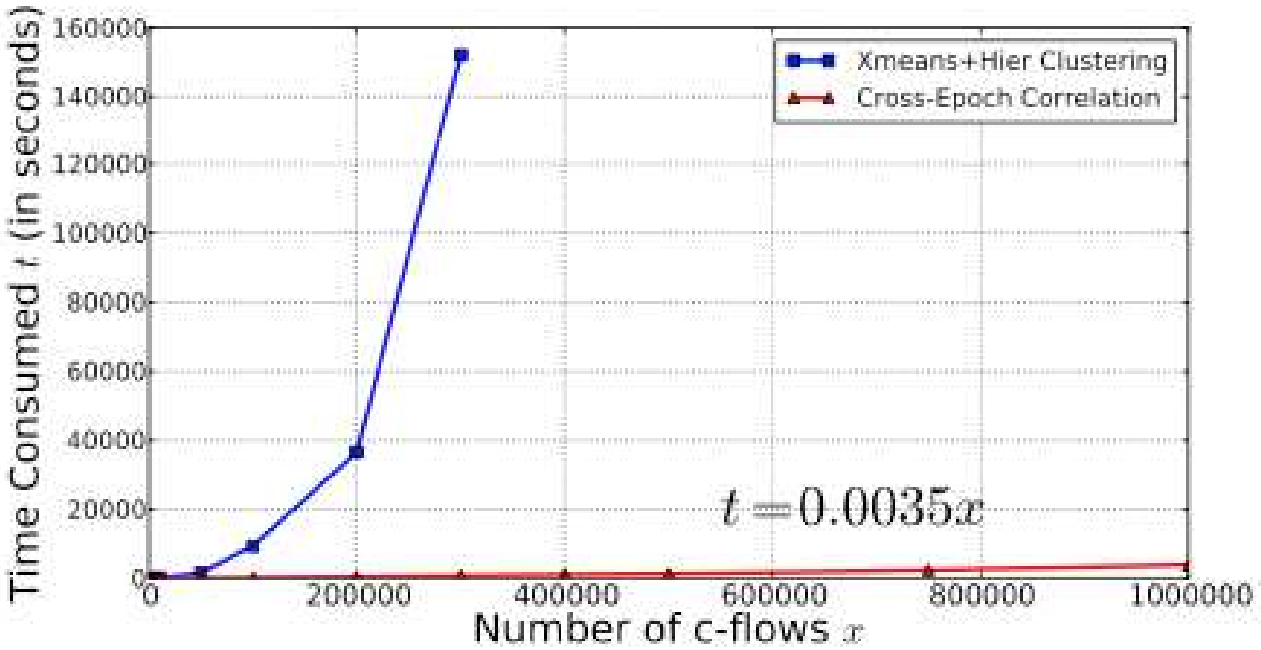
SR_T	For each Per_{Exp}, TP(bots/23), FP(noises/1460)									
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.01	48%, 0.1%	83%, 0.5%	96%, 1%	96%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 6%	100%, 8%
0.025	52%, 0%	87%, 0.5%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 7%	100%, 8%
0.05	48%, 0.1%	100%, 0.3%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 5%	100%, 7%	100%, 7%
0.075	48%, 0.2%	100%, 0.3%	100%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 6%	100%, 7%	100%, 8%
0.1	39%, 0.3%	78%, 0.8%	100%, 1%	100%, 2%	100%, 3%	100%, 3%	100%, 5%	100%, 5%	100%, 7%	100%, 8%
1	30%, 0.5%	65%, 0.8%	96%, 1%	100%, 2%	100%, 3%	100%, 4%	100%, 5%	100%, 5%	100%, 7%	100%, 8%

Table 4: Detection Rates of Cross-Epoch Correlation using B-Sampling

- Cross-epoch correlation together with B-Sampling can detect all the bots for most of the combinations of SR_T (target sampling rate) and Per_{Exp} (expected percentage of suspicious hosts)

Evaluation

- Cross-Epoch Correlation
  - Time consumption of cross-epoch correlation compared to BotMiner's clustering algorithm (X-means + hierarchical clustering)



- Cross-epoch correlation has great scalability

Evaluation

- Fine-Grained Detectors
  - Detection Results

SR_T	For each Per_{Exp}, TP(bots/23), FP(noises/1460)									
	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.01	48%, 0	83%, 0	96%, 0	96%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.025	52%, 0	87%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.05	48%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.075	48%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
0.1	39%, 0	78%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0
1	30%, 0	65%, 0	96%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0	100%, 0

- Eliminate all the false positives
- Achieve high detection

	With Flow-Corr ($Per_E = 5\%, M = 3$)						direct
SR_T	0.01	0.025	0.05	0.075	0.1	1	
Per of Pkts	1.7%	2.9%	2.1%	3%	4.3%	2%	100%
Time	33s	39s	35s	40s	49s	33s	858s

with our approach

direct deployment

- Fine-grained detectors only need to investigated less than 5% traffic and use much less time.

Discussion

- High-Speed Networks
  - Given 2 hr process time of cross-epoch correlation and t=0.0035 * "# of c-flows" , our system can process 2M c-flows (i.e., "# of c-flows")
  - College network: 200K c-flows extracted from 200Mbps traffic
  - 2M c-flows would result from 2Gbps, indicating that the cross-epoch correlation can be used in 2 Gbps networks
- Evasion
  - Randomize communication patterns to decrease the packet sampling rates and evade cross-epoch correlation

Conclusion

- A botnet-aware adaptive sampling algorithm
  - Keep the actual packet sampling rate close to the target sampling rate
  - High sampling rates for botnet C&C related packets compared
- Cross-epoch correlation
  - Effectively and efficiently identify bots by investigating their persistently similar communication patterns
- A new botnet detection system employing layered traffic analysis approach

Thanks!
Questions?