Traffic Aggregation for Malware Detection

Ting-Fang Yen

Carnegie Mellon University,
Pittsburgh, Pennsylvania
U.S.A.

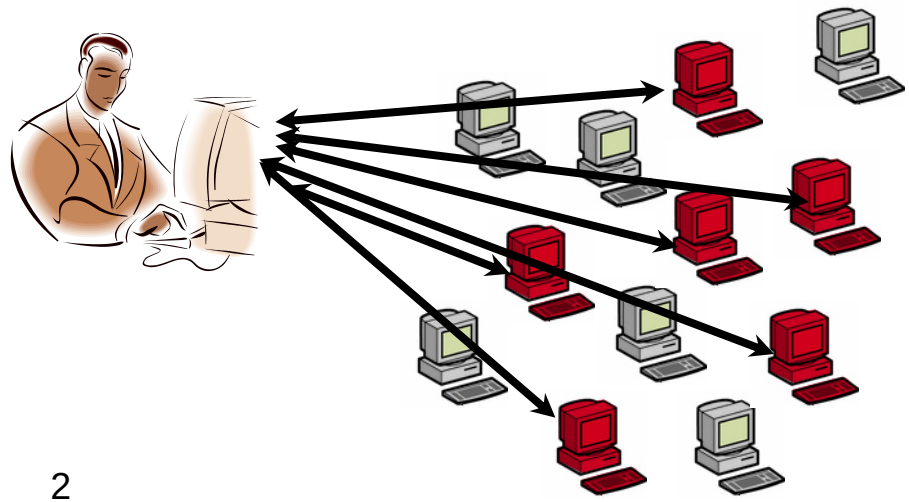
Michael K. Reiter

University of North Carolina
at Chapel Hill,
Chapel Hill, North Carolina
U.S.A.

DIMVA 2008

Background

- Stealthy malware: spyware, adware, bots, ….
  - Subtle command/control system
  - Organized malicious activities
    - Spamming, hosting phishing sites, DDoS attacks



Traffic Aggregation for Malware Detection (TAMD)

- Observe flow records at network border
- Assumptions:
  - More than one infected host in the network
  - Malware communication patterns different from benign hosts
- Traffic aggregates: network traffic sharing common characteristics
  - Question: what characteristics can identify malware?

Aggregate Characteristics

- Common destination
  - Spyware “phone-home”, botnet controller, bot update server, DDoS attack victim
- Similar Payload
  - Bot commands
- Similar platform
  - Platform-dependent infections
- Challenge: identify malware traffic while limiting the number of false alarms

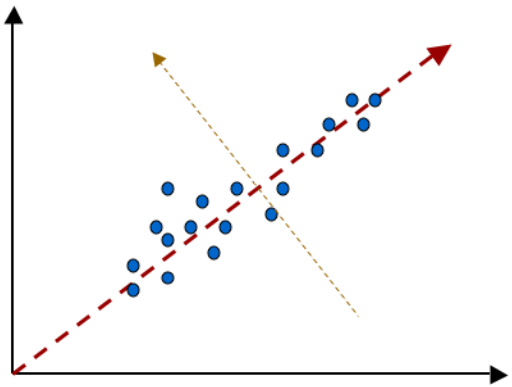
Destination Aggregates

- Internal hosts contacting the same “busier-than-usual” external subnets
  - Use past traffic as baseline
- Represent internal hosts as vectors
  - Dimensions (i.e., D_1, D_2, ...) correspond to external subnets

	D_1	D_2	D_3	D_4	D_5
H_1 = <	1,	1,	0,	1,	1 >
H_2 = <	1,	1,	1,	0,	0 >
H_3 = <	1,	1,	0,	1,	0 >

Destination Aggregates (cont'd)

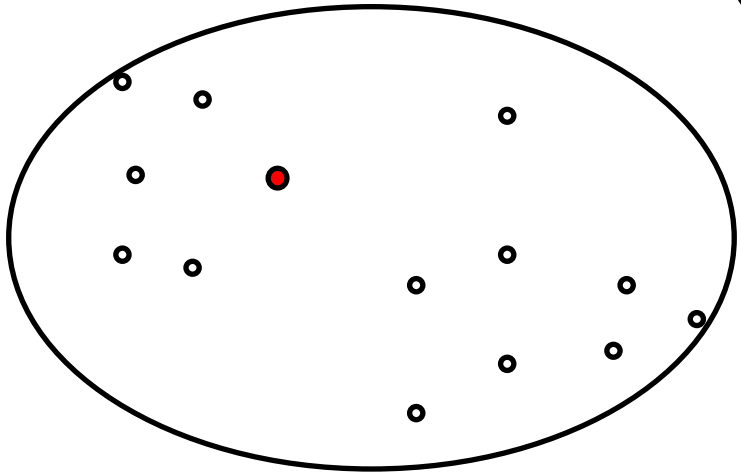
- Dimension Reduction
  - Principal Component Analysis (PCA)
  - Re-interpret data with new axes that captures most of the data variance



- Clustering
  - Iteratively select furthest vector to be new hub
  - Clusters contain hosts contacting the same "busier-than-usual" subnets

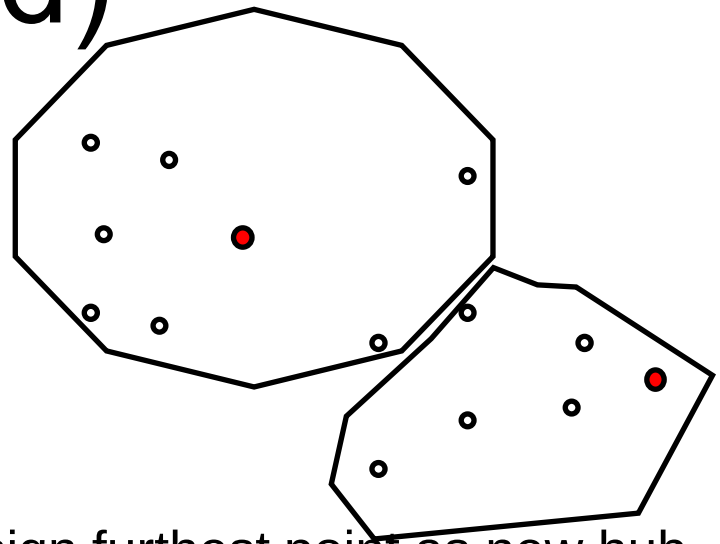
Destination Aggregates (cont'd)

1



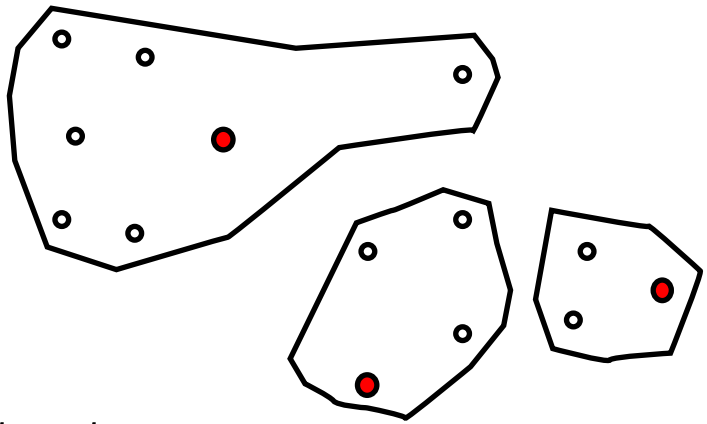
Assign random point as initial hub.

2



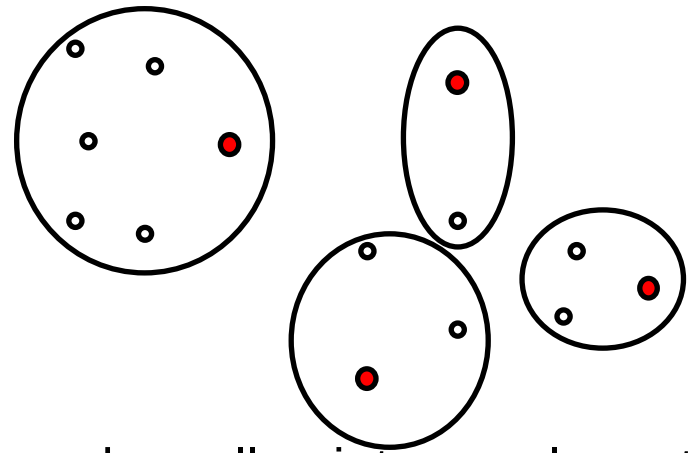
Assign furthest point as new hub. Re-cluster.

3



Iterate.

4



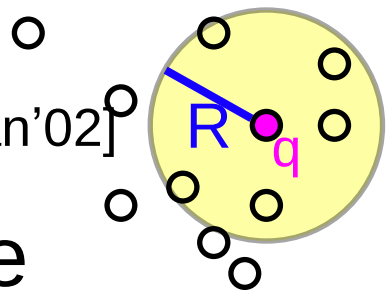
Stop when all points are closer to their hub than half of the average hub-hub distance.

Payload Aggregates

- Flows with “similar” payload prefix
- Edit distance as similarity metric
  - Number of character insertions, deletions, substitutions, to turn one string into the other
  - Captures syntactic similarities
    - “.bot.execute 1 notepad.exe”
      “.bot.execute 0 cmd.exe”
      “abcdeeeeeenoopttuxx1.. . ”
  - However, computationally expensive

Payload Aggregates (cont'd)

- Locality Sensitive Hashing [Datar-Immorlica-Indyk-Mirrokni'04]
  - Near-neighbor search: close points hash to same buckets
- Edit Sensitive Parsing [Cormode-Muthukrishnan'02]
  - Embed edit distance into L1 distance



- As a result...
  - Only compute edit distance for strings whose vectors hash to same buckets
  - Time roughly proportional to size of data set

Platform Aggregates

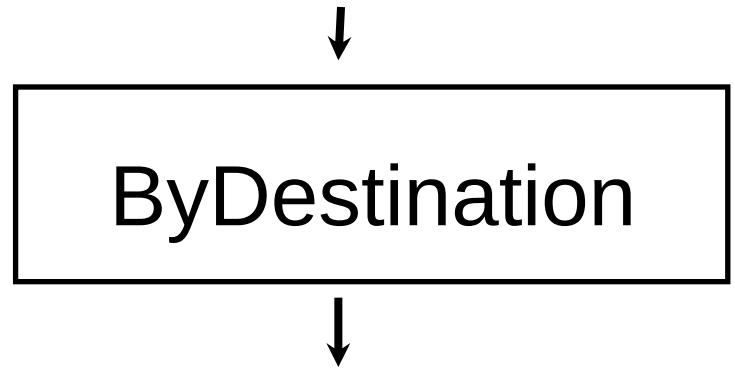
- Traffic from hosts of similar platform
  - TTL (Time-to-Live) field
  - Communication with characteristic sites
    - e.g., Microsoft time server

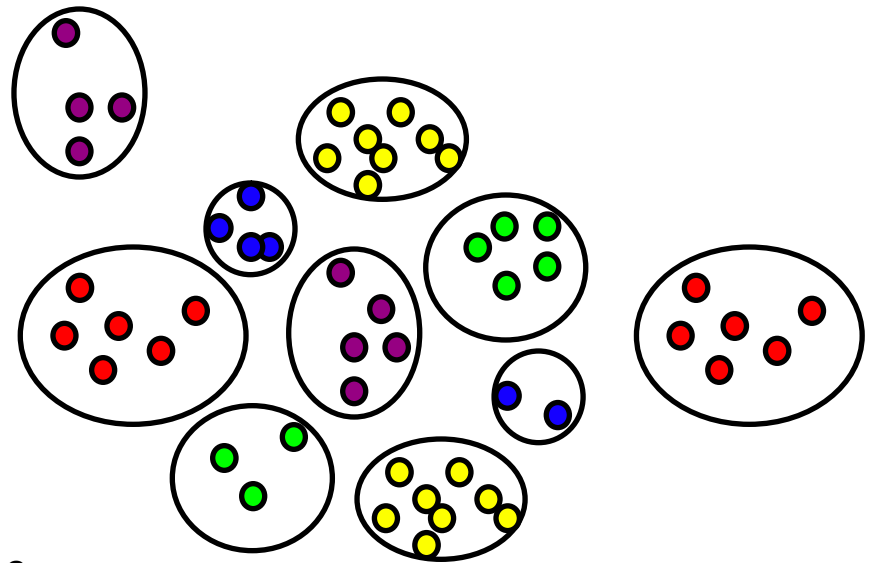
Multi-Level Aggregation

- Aggregation Functions:
  - ByDestination
  - ByPayload
  - ByPlatform
- In combination, refine resulting aggregates
  - Traffic sharing multiple relevant characteristics
  - Example: platform-dependent infections that contact common sites

Aggregation Example

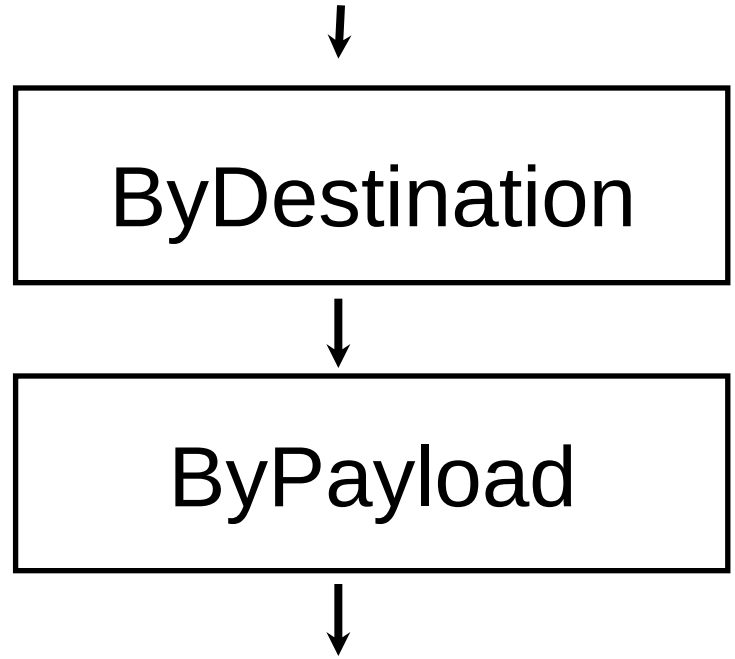
Multiple infected hosts contacting sites uncommon to benign hosts.

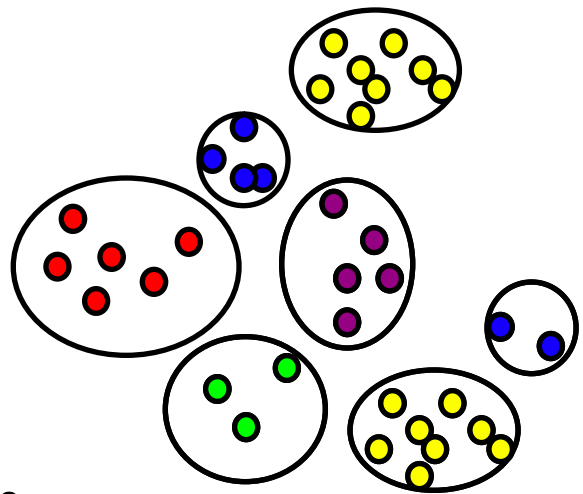




Aggregation Example (cont'd)

Multiple infected hosts contacting sites uncommon to benign hosts.
Malware communication similar among infected hosts.

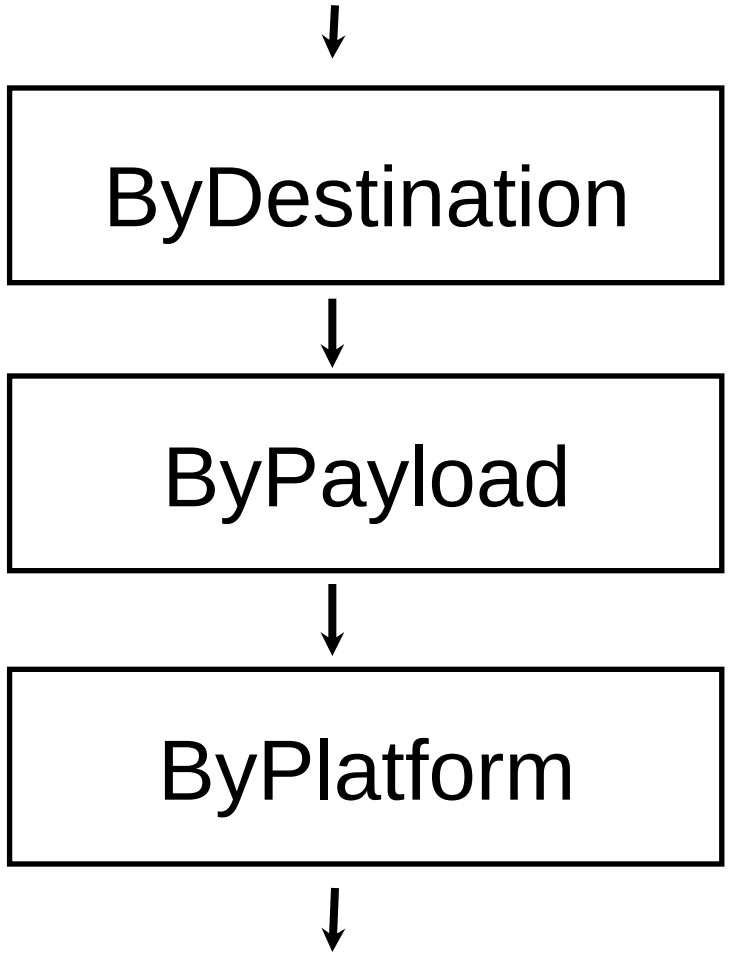


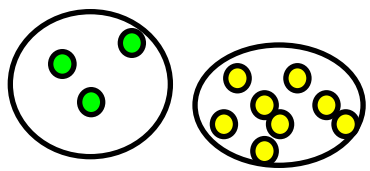


Aggregation Example (cont'd)

Multiple infected hosts contacting sites uncommon to benign hosts.
Malware communication similar among infected hosts.

Platform-dependent infection.





Evaluation Data

- Network traces from Carnegie Mellon University network border
  - Two /16 subnets, over 33,000 hosts
  - Argus flow records:

IP Header	Transport Header	Flow Attribute
Source IP	Source Port	Byte Count
Destination IP	Destination Port	Packet Count
Protocol	TCP Sequence Number	Payload (64 bytes)
TTL	TCP Window Size	

    - Captures ~5000 flows/sec
    - 9 a.m. to 3 p.m. daily
  - Experiments use TCP and UDP traffic only

Evaluation Data (cont'd)

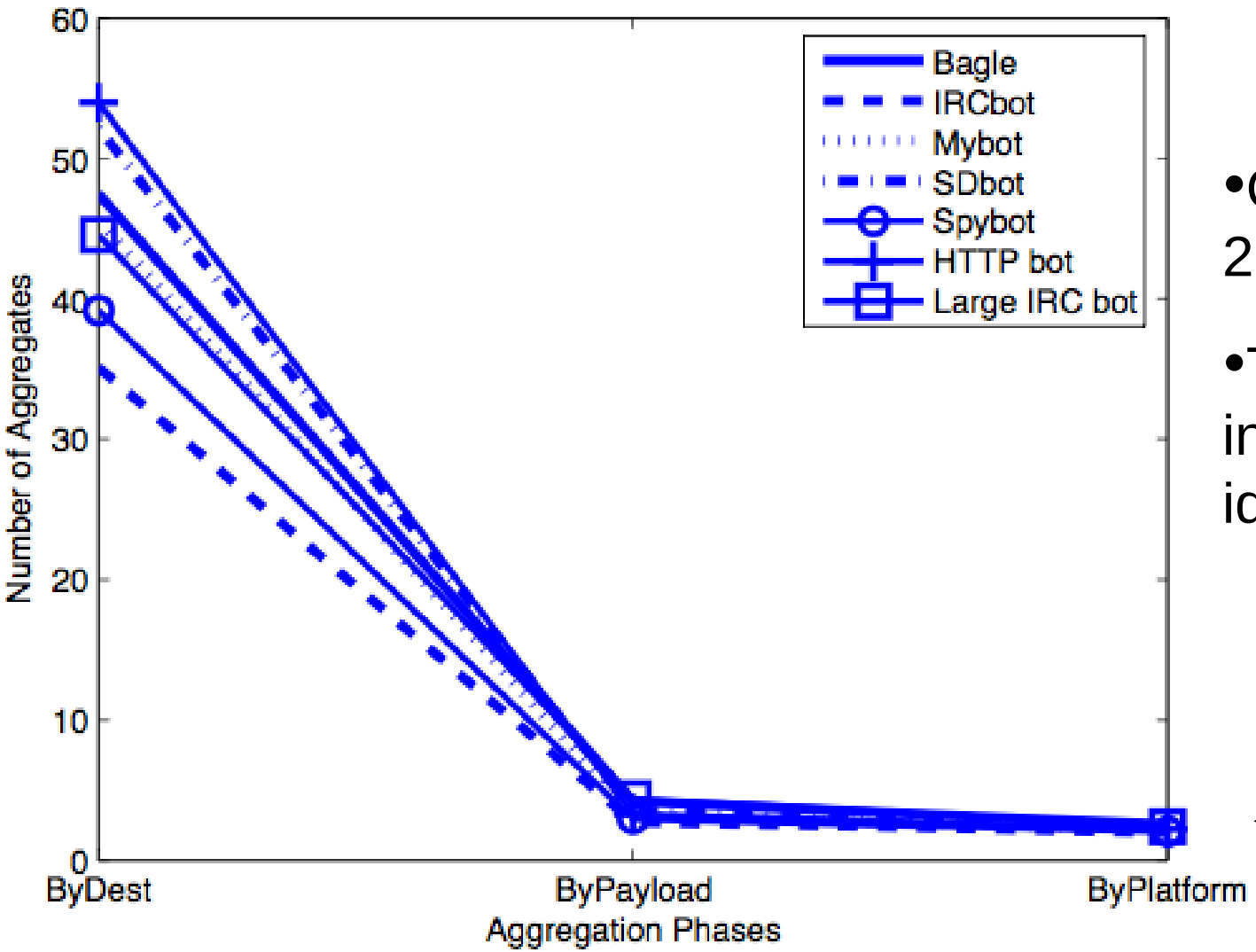
- Network traces from malware in virtual machines
  - Bagle, IRCbot, Mybot, SDbot
  - Infect 3~8 Windows XP virtual hosts with each malware binary
  - One hour of traffic from each malware
- Network traces from botnets in honeynets
  - Spybot : Four bots, 32-minute trace
  - HTTP-bot : Four bots, three-hour trace
  - Large botnet : > 340 bots, seven-minute trace

Evaluation

- For every hour of campus traffic,
  - For every malware,
    - Assign malware traffic to randomly selected internal hosts of same platform
      - Comprise 0.0097% of all internal hosts
    - Input to aggregation functions

- Repeat over every hour during three weeks in November/December 2007

Results

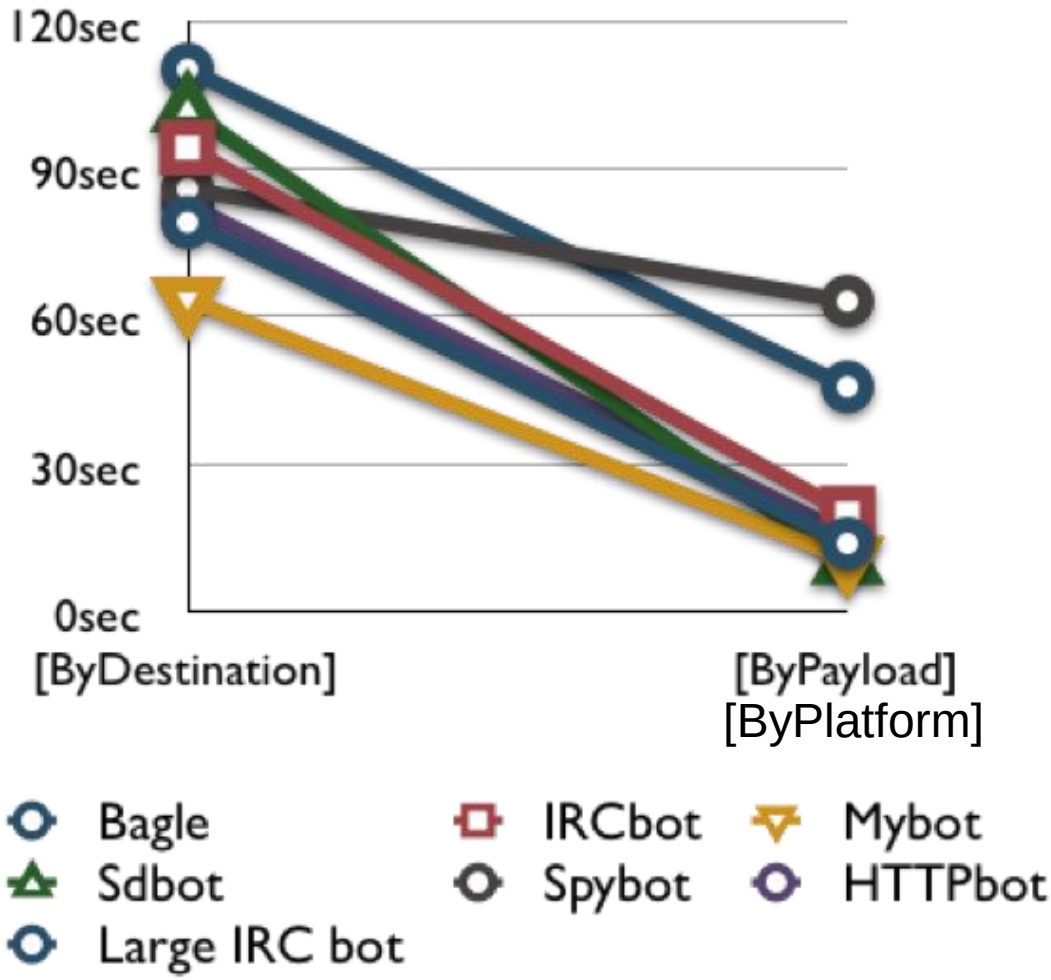


- On average, identified 2.23 aggregates
- The single aggregate c infected hosts is always identified

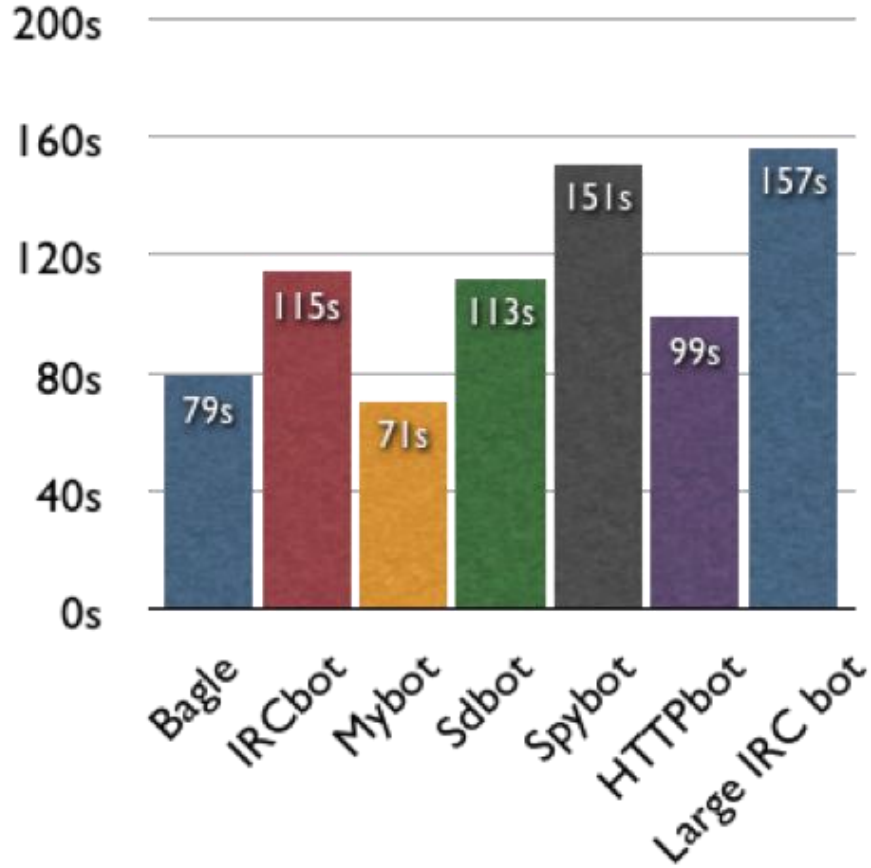
←2.23

Performance Statistics

Function Run Time



Total Run Time



Alternative Botnet Architectures

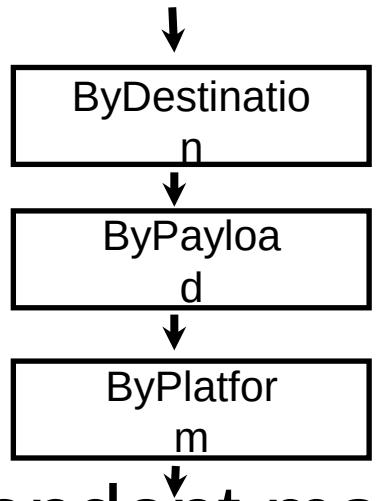
- Peer-to-peer (P2P):
  - Hard-coded peer list
  - Bots report back to designated site
  - Use P2P to transfer URLs for downloading binaries
- Hybrid: Smaller centralized botnets peer in P2P

Limitations and Ongoing Work

- Temporal locality in malware communication
  - But sparse communication restricts botnet size and responsiveness
- Diversity in hosts' platforms
  - Good results with only ByDestination and ByPayload
- P2P with peer discovery through random probing
  - ByPayload or ByPlatform
- Encrypted payload
  - Extend "similar" to include encrypted traffic
- Isolated bots

Conclusion

- Traffic Aggregation for Malware Detection (TAMD): Identifies traffic sharing common network characteristics
  - Common destination
  - Similar payload
  - Common platform



- Detects stealthy platform-dependent malware contacting common sites
  - Successful even when number of simulated infected hosts comprise 0.0097% of internal hosts