BLINC: Multilevel Traffic Classification in the Dark

Thomas Karagiannis, UC Riverside
Konstantina Papagiannaki, Intel Research Cambridge
Michalis Faloutsos, UC Riverside
The problem of workload characterization

• The goal: Classify Internet traffic flows according to the applications that generate them “in the dark”
  – No port numbers
  – No payload
The problem of workload characterization – Why in the dark?

• Traffic profiling based on TCP/UDP ports
  – Misleading

• Payload-based classification
  – Practically infeasible

• Applications are “hiding” their traffic
  – P2P applications, skype, etc.

• Recent research approaches
  – Statistical/machine-learning based classification (Roughan et al. IMC’04, Moore et al. SIGMETRICS’05)
  – Sensitive to network dynamics such as congestion
Our contributions

• We present BLINC (BLINd Classification), a fundamentally different “in the dark” approach
  – We shift the focus to the Internet host
  – We analyze host behavior at three levels
    • Social
    • Functional
    • Application

• We identify “signature” communication patterns

• Highly accurate classification
Outline

• Developing a classification benchmark
  – Payload-based classification

• BLINC design
  – Multilevel classification
  – Signature communication patterns

• BLINC evaluation
Classification benchmark

• Packet-traces with machine readable headers
  – Residential (2 traces)
    • 25 hours & 34 hours, 110 Mbps
    • web (35%), p2p (32%)
  – Genome campus
    • 44 hours, 25 Mbps, ftp (67%)

• Classification based on payload signatures
  – Caveats: Nonpayload (1%-2%), Unknown (6%-16%)
BLINC overview

• In the dark classification
  – No examination of port numbers
  – No examination of user payload

• Characterize the host
  – Insensitive to congestion and path changes

• Deployable with existing equipment
  – Operates on flow records
BLINC: Classification process

• Characterize the host
  – Social : Popularity/Communities
  – Functional : Consumer/provider of services
  – Application : Transport layer interactions

• Identify signature communication patterns

• Match observed behavior to signatures
1. Social level

• Characterization of the popularity of hosts

• Two types of behavior:
  – Based on number of destination IPs
  – Communities: Groups of communicating hosts
1. Social level: Popularity

- Reveals only basic application traffic properties

Heavier tail of CCDF of destination IPs for P2P and malware
1. Social level: Communities

- Communication cliques
- Perfect cliques
  - Attacks
- Partial cliques
  - Collaborative applications (p2p, games)
- Partial cliques with same domain IPs
  - Server farms (e.g., web, dns, mail)
2. Functional level

- We characterize based on tuple (IP, Port)

- We identify three types of behavior
  - Client: Consumer of services
  - Server: Provider of services
  - Collaborative
2. Functional level: Client vs. Server

Observation:
The host uses a different ephemeral src port for every flow

Rule:
Hosts that use a large number of source ports are clients
2. Functional level: Client vs. Server

Rule:
Hosts that use a small number of source ports are offering services on these ports

Observation:
The host uses only two src ports for all flows

src port: 80
src port: 80
src port: 443
2. Functional level: Characterizing the host

flows vs. source ports per application

Collaborative applications: No distinction between servers and clients

Obscure behavior due to multiple mail protocols and passive ftp
3. Application level

• Interactions between network hosts display diverse patterns across application types.

• We capture patterns using “graphlets”
  – Target most typical behavior
  – Relationship between fields of the 4-tuple
3. Application level: Graphlets

- Graphlets have four columns corresponding to the 4-tuple: src IP, dst IP, src port and dst port
- Each node is a distinct entry for each column
- Lines connect nodes when flows contain the specific field values

<table>
<thead>
<tr>
<th>sourceIP</th>
<th>destinationIP</th>
<th>sourcePort</th>
<th>destinationPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.1.1</td>
<td>10.0.0.0</td>
<td>1026</td>
<td>135</td>
</tr>
</tbody>
</table>

![Graphlet Diagram]

Intel Research
### 3. Graphlet Generation (FTP)

<table>
<thead>
<tr>
<th>sourceIP</th>
<th>destinationIP</th>
<th>sourcePort</th>
<th>destinationPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>21</td>
<td>10001</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>20</td>
<td>10002</td>
</tr>
<tr>
<td>X</td>
<td>Z</td>
<td>21</td>
<td>3000</td>
</tr>
<tr>
<td>X</td>
<td>Z</td>
<td>1026</td>
<td>3001</td>
</tr>
</tbody>
</table>

![Graph Representation](image)
3. Graphlet Library
Heuristics: Further improving performance

- Using the transport layer protocol.
Heuristics: Further improving performance

• Using the relative cardinality of sets.

Cardinality of set of dst IPs versus set of dst ports varies with the application
Heuristics: Further improving performance

• Using the relative cardinality of sets.

WEB: #dst ports >> # dst IPs
P2P: #dst ports <= # dst IPs
Heuristics: Further improving performance

- Using the communities

Known: WEB

10.0.0.0

10.0.0.1

Probably WEB too!!
Heuristics: Further improving performance

• Other heuristics:
  – Using the per-flow average packet size
  – Recursive (mail/dns servers talk to mail/dns servers, etc.)
  – Failed flows (malware, p2p)
Classification Results

• We evaluate BLINC using two metrics:
  – Completeness
    • Percentage classified by BLINC
  – Accuracy
    • Percentage classified by BLINC correctly

• We compare against payload classification
  – Exclude unknown and nonpayload flows
BLINC achieves highly accurate classification

80%-90% completeness!
>90% accuracy!!
Characterizing the unknown: Non-payload flows

BLINC is not limited by non-payload flows or unknown signatures

Flows classified as attacks reveal known exploits
BLINC issues and limitations

• Extensibility
  – Creating and incorporating new graphlets
• Application sub-types
  – e.g., BitTorrent vs. Kazaa
• Transport-layer encryption
  – then what?
• NATS
  – Should handle most cases
• Access vs. Backbone networks?
  – Should handle but no data to test
Conclusions

• A new way of thinking of the classification problem
  – Classify nodes instead of flows
  – Multi-level analysis:
    • social, functional, transport-layer characteristics
    • each level provides corroborative evidence or insight

• BLINC works well in practice
  – classifies 80-90% of the traffic
  – with >90% accuracy

• Going beyond payload-based classification
  – Nonpayload/unknown flows

• Building block for security applications