PeerRush
Mining for Unwanted P2P Traffic

Babak Rahbarinia\textsuperscript{a}, Roberto Perdisci\textsuperscript{a,\ b}, Andrea Lanzi\textsuperscript{c}, Kang Li\textsuperscript{a}

\textsuperscript{a}University of Georgia
\textsuperscript{b}Georgia Tech
\textsuperscript{c}EURECOM
Introduction

• P2P traffic represents a significant fraction of all Internet traffic
  — Apps: File Sharing, VoIP, P2P Botnets, ...

• Net admins need to categorize traffic that crosses their network’s perimeter
  — Detect malware infections related to P2P botnets
  — Identify/block some types of P2P traffic

• Identifying P2P traffic can aid Net-based IDSES
Previous Work

• Several papers on P2P traffic detection
  – Port numbers, Sig-based, DPI, statistical traffic analysis
• Very little research on non-sig-based P2P traffic categorization
  – Profiling P2P traffic (Hu et al., Computer Networks’09)
  – only applied to non-encrypted traffic, very few apps
• Some work on P2P botnet detection
  – BotMiner (Gu et al.), Statistical traffic fingerprints
    (Zhang et al.), Traders or Plotters? (Yen et al.), …
  – Cannot distinguish between different P2P botnets
PeerRush Goals

• Detect and *categorize P2P traffic*
  – Generic/flexible traffic categorization approach
  – Statistical traffic features
  – Agnostic to payload encryption

• Identify *unwanted* P2P traffic
  – “unwanted” depends on network management and security policies
  – Includes malicious traffic, such as P2P botnets
  – May include other legit but unwanted apps, such as file sharing (eMule, BitTorrent, etc.)
PeerRush: System Overview

1. P2P host detection
   - P2P traffic samples
   - Non-P2P samples

2. P2P traffic categorization
   - Application profile 1
   - Application profile 2
   - Application profile 3
   - Application profile N

Auxiliary P2P traffic disambiguation

P2P traffic categorization reports
P2P Host Detection - Overview

- **Input**: live network traffic
- **Approach**: statistical two-class classifier
- **Output**: IPs that generate P2P traffic
P2P Host Detection - Features

$[f_1, f_2, ..., f_k]$ 

- **Statistical features**
  - # TCP/UDP “connections” with no DNS query
  - # failed connections (peer churn effect)
  - Non-DNS dst IPs scattered in many different networks
    - successful, failed, and all connections

- **Non-P2P traffic has low feature values**
  - e.g., web traffic
  - Most non-P2P connections “start” with DNS query
  - Only few failed connections
P2P Traffic Categorization - Overview

- **Input:**
  - traffic from each P2P host
  - *P2P management flows*

- **Approach:**
  - Application profiles modeled by one-class classifiers

- **Output:**
  - P2P traffic profile matches
P2P Traffic Categorization - Features

• Different P2P apps generate different traffic
  – Use different P2P protocols
  – Connect to different network of peers

• P2P management (or control) flows
  – P2P traffic overall depends on user activities
  – need to find user-independent features!
  – better to focus on P2P control traffic
    • e.g., periodic “keep alive” messages
    • protocol-specific, more user-independent

• 1\textsuperscript{st} goal
  – separate management flows from data flows
Finding Management Flows

• Heuristics-based approach
  1. Consider only non-DNS flows
  2. Consider long-lived (TCP/UDP) flows
     • packet exchange for a significant portion of analysis window
  3. Leverage inter-packet delays
     • Data transfers typically involve bursts of packets
     • Management messages are exchanged periodically
Management Flow Features

- Distribution of bytes per packet (BPP)
- Distribution of inter-packed delays (IPD)
  - Find top $n$ BPP and IPD peaks
  - Measure peak location and relative height
P2P App Profiles

- One-class classification approach
  - Each traffic profile trained using only examples of traffic from target app
  - Flexibility: different decision function and threshold per each app

\[
\{f_1, f_2, ..., f_m\} \rightarrow \text{P2P traffic categorization}
\]

\[
\begin{align*}
\text{dist}_1&(\text{score}_1, \theta_1) \\
\text{dist}_2&(\text{score}_2, \theta_2) \\
\text{dist}_3&(\text{score}_3, \theta_3) \\
\text{dist}_N&(\text{score}_N, \theta_N)
\end{align*}
\]

- One match
- Multiple matches (need disambiguation)
- Unknown P2P app
Evaluation Datasets

- 5 ordinary (non-malicious) apps
  - Several days per app
  - Hundreds of GB of traffic
Evaluation Datasets

• Traffic from 3 real-world P2P botnets
  – Storm, Waledac, Zeus P2P (encrypted)

• Non-P2P traffic
  – about 5 days of CS dept. network
  – custom sniffing, anonymizes packets “on the fly”
  – pruned all src IPs that are suspected P2P hosts
    • any query to *.skype.com, any match of Snort P2P rules
    • 21 out of 931 hosts pruned overall
Eval of P2P Host Detection

• Cross-validation on non-malicious apps
  – Datasets: ordinary P2P traffic + non-P2P traffic
  – Classifier: Boosted Decision Trees

<table>
<thead>
<tr>
<th>time window</th>
<th>TP</th>
<th>FP</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>60 min</td>
<td>99.5%</td>
<td>0.1%</td>
<td>1</td>
</tr>
<tr>
<td>40 min</td>
<td>99.1%</td>
<td>0.8%</td>
<td>0.999</td>
</tr>
<tr>
<td>20 min</td>
<td>98.4%</td>
<td>1.1%</td>
<td>0.999</td>
</tr>
<tr>
<td>10 min</td>
<td>97.9%</td>
<td>1.2%</td>
<td>0.997</td>
</tr>
</tbody>
</table>

• Separate “hold-out” test on P2P Botnets

<table>
<thead>
<tr>
<th>Time Win.</th>
<th>Botnet</th>
<th>Instances</th>
<th>TPs</th>
<th>IPs detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>60 min</td>
<td>Storm</td>
<td>306</td>
<td>100%</td>
<td>13 out of 13</td>
</tr>
<tr>
<td></td>
<td>Zeus</td>
<td>825</td>
<td>92.48%</td>
<td>1 out 1</td>
</tr>
<tr>
<td></td>
<td>Waledac</td>
<td>75</td>
<td>100%</td>
<td>3 out 3</td>
</tr>
<tr>
<td>10 min</td>
<td>Storm</td>
<td>1,834</td>
<td>100%</td>
<td>13 out of 13</td>
</tr>
<tr>
<td></td>
<td>Zeus</td>
<td>4,877</td>
<td>33.46%</td>
<td>1 out 1</td>
</tr>
<tr>
<td></td>
<td>Waledac</td>
<td>444</td>
<td>100%</td>
<td>3 out 3</td>
</tr>
</tbody>
</table>
Eval of P2P Categorization

- App profile = one-class classifier
  - Different “optimum” classifier configuration per app
  - Cross-validation results

<table>
<thead>
<tr>
<th>App.</th>
<th>#Inst.</th>
<th>Configuration</th>
<th>TP</th>
<th>FP</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skype</td>
<td>526</td>
<td>60min; KNN; 32 feat.; PCA</td>
<td>96.54%</td>
<td>0.74%</td>
<td>0.998</td>
</tr>
<tr>
<td></td>
<td>579</td>
<td>10min; Parzen; 16 feat.; -</td>
<td>91.27%</td>
<td>1.00%</td>
<td>0.978</td>
</tr>
<tr>
<td>eMule</td>
<td>387</td>
<td>60min; Parzen; 16 feat.; Scal.</td>
<td>90.64%</td>
<td>0.92%</td>
<td>0.989</td>
</tr>
<tr>
<td></td>
<td>483</td>
<td>10min; KNN; 8 feat.; PCA</td>
<td>88.40%</td>
<td>1.16%</td>
<td>0.961</td>
</tr>
<tr>
<td>Frostwire</td>
<td>382</td>
<td>60min; KNN; 12 feat.; PCA</td>
<td>85.58%</td>
<td>0.96%</td>
<td>0.966</td>
</tr>
<tr>
<td></td>
<td>467</td>
<td>10min; KNN; 8 feat.; PCA</td>
<td>92.68%</td>
<td>1.25%</td>
<td>0.989</td>
</tr>
<tr>
<td>µTorrent</td>
<td>370</td>
<td>60min; KNN; 8 feat.; -</td>
<td>92.94%</td>
<td>1.30%</td>
<td>0.948</td>
</tr>
<tr>
<td></td>
<td>609</td>
<td>10min; Parzen; 4 feat.; Scal.</td>
<td>94.55%</td>
<td>1.24%</td>
<td>0.992</td>
</tr>
<tr>
<td>Vuze</td>
<td>376</td>
<td>60min; KNN; 8 feat.; -</td>
<td>91.92%</td>
<td>0.95%</td>
<td>0.979</td>
</tr>
<tr>
<td></td>
<td>514</td>
<td>10min; KNN; 8 feat.; PCA</td>
<td>84.18%</td>
<td>1.17%</td>
<td>0.964</td>
</tr>
<tr>
<td>Storm</td>
<td>162</td>
<td>60min; Parzen; 16 feat.; -</td>
<td>100%</td>
<td>0%</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>391</td>
<td>10min; Parzen; 12 feat.; PCA</td>
<td>100%</td>
<td>0%</td>
<td>1.000</td>
</tr>
<tr>
<td>Zeus</td>
<td>375</td>
<td>60min; KNN; 4 feat.; -</td>
<td>97.29%</td>
<td>0.99%</td>
<td>0.996</td>
</tr>
<tr>
<td></td>
<td>188</td>
<td>10min; KNN;12 feat.; -</td>
<td>94.53%</td>
<td>0.79%</td>
<td>0.976</td>
</tr>
<tr>
<td>Waledac</td>
<td>37</td>
<td>60min; Gaussian; 12 feat.; PCA</td>
<td>99.99%</td>
<td>0.90%</td>
<td>0.998</td>
</tr>
</tbody>
</table>
## Overall Eval of P2P Categorization

<table>
<thead>
<tr>
<th>Application</th>
<th>TP</th>
<th>FP</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skype</td>
<td>100%</td>
<td>0.86%</td>
<td>1</td>
</tr>
<tr>
<td>eMule</td>
<td>93.59%</td>
<td>1.44%</td>
<td>0.9968</td>
</tr>
<tr>
<td>Frostwire</td>
<td>88.31%</td>
<td>0.97%</td>
<td>0.9873</td>
</tr>
<tr>
<td>μTorrent</td>
<td>96.97%</td>
<td>1%</td>
<td>0.9789</td>
</tr>
<tr>
<td>Vuze</td>
<td>93.1%</td>
<td>0.7%</td>
<td>0.9938</td>
</tr>
<tr>
<td>Storm</td>
<td>100%</td>
<td>0%</td>
<td>1</td>
</tr>
<tr>
<td>Zeus</td>
<td>96.69%</td>
<td>1.26%</td>
<td>0.9964</td>
</tr>
<tr>
<td>Waledac</td>
<td>57.14%</td>
<td>0.83%</td>
<td>0.9420</td>
</tr>
</tbody>
</table>

**time window: 60 minutes**

- Classified as “unknown”: 3.96% (29 out of 732)
- Misclassified as other P2P: 0% (0 out of 732)
- Disambiguation needed: 4.64% (34 out of 732)
  - Correctly disambiguated: 33, Incorrectly disambiguated: 1

**Total misclassified as other P2P: 0.14% (1 out of 732)**
Conclusion

- PeerRush allows for flexible and accurate P2P traffic detection and categorization
- Enables detection of unwanted P2P traffic
  - different types of modern P2P botnets
  - unwanted “ordinary” P2P apps
  - agnostic to traffic encryption
- Extensive evaluation
  - 5 ordinary P2P apps + 3 modern P2P botnets
  - High accuracy of different system components
  - Promising results on robustness against traffic noise (results in the paper)
perdisci@cs.uga.edu
Overall Eval of P2P Categorization

<table>
<thead>
<tr>
<th>Application</th>
<th>TP</th>
<th>FP</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skype</td>
<td>90.4%</td>
<td>1.29%</td>
<td>0.9891</td>
</tr>
<tr>
<td>eMule</td>
<td>94.87%</td>
<td>2.39%</td>
<td>0.9935</td>
</tr>
<tr>
<td>Frostwire</td>
<td>94.73%</td>
<td>0.48%</td>
<td>0.9927</td>
</tr>
<tr>
<td>μTorrent</td>
<td>98.99%</td>
<td>0.66%</td>
<td>0.9997</td>
</tr>
<tr>
<td>Vuze</td>
<td>93.22%</td>
<td>3.02%</td>
<td>0.9873</td>
</tr>
<tr>
<td>Storm</td>
<td>45.45%</td>
<td>0%</td>
<td>0.7273</td>
</tr>
<tr>
<td>Zeus</td>
<td>97.32%</td>
<td>0.72%</td>
<td>0.9991</td>
</tr>
<tr>
<td>Waledac</td>
<td>40%</td>
<td>0.8%</td>
<td>0.8610</td>
</tr>
</tbody>
</table>

Time window: 60 minutes

* Classified as “unknown”: 6.15% (45 out of 732)
* Misclassified as other P2P: 0.68% (5 out of 732)
* Disambiguation needed: 4.37% (32 out of 732)
  - Correctly disambiguated: 32, Incorrectly disambiguated: 0

Total misclassified as other P2P: 0.68% (5 out of 732)
Evaluation Datasets

• 5 non-malicious P2P apps
  – Skype
  – μTorrent
  – eMule
  – Vuze
  – Frostwire

• 3 P2P botnets
  – Storm
  – Waledac
  – Zeus P2P