Intrusion as (Anti)social Communication: Characterization and Detection

Natallia V. Katenka
(Joint work with Q. Ding, P. Barford, E. Kolaczyk, and M. Crovella)
Supported by ONR and NSF

Group 5: Jiayong Li, Yao Lin, Hetong Li
Outline

• Motivation
• Data Description
• Network Representation
• Community Detection
• Detecting Malicious Source
• Experiment
  -Result: Analysis of 1 Day
  -Result: Analysis of 14 Days
• Conclusion
Motivation

• Network-based intrusion detection systems play a significant role in protecting IT assets from malicious attacks.

• Standard (signature-based) intrusion detection systems fail against previously unseen attacks, since:

  – these systems rely on previously seen header/content patterns of malicious/unwanted traffic.
Motivation

• Anomaly-based network attack detection offers a compelling alternative to signature-based methods.

• **Question:** How to define essence of ‘normal’ and ‘abnormal’ communication behavior? **Focus on:** communication networks for community and intrusion detection.

Main Hypothesis: if intrusion is entering a community to which one does not belong, then in a network, intrusion attempts may be detected by looking for communication that does not respect community boundaries.
Motivation

• Hypothesis: Defines intrusion in terms of locations in social space (where normality often has well-understood properties).
• Bipartite graph: a standard representation of communication patterns in network traffic.
• Cut-vertex: Cut a node from a graph.
Motivation

In the graph below, $C$ is a cut-vertex.

The edges $AC$, $BC$, $CD$, $CF$ are the edges which would be removed if $C$ were cut. The graph would be separated into the two components $AB$ and $DEF$. 
Data Description

- To create and characterize networks of communication patterns, we use netflow data collected from a large European Internet Service Provider (May 1, 2007 - May 14, 2007).

- Each netflow record consists of:
  - source IP, destination IP, source Port, destination Port, start/end time, number of bytes and packets in sent message,
  - BUT no actual message content, so signature-based methods CANNOT be applied.
Data Description

- To label malicious sources in our data in order to validate our intrusion detection method, we use Firewall logs provided by Dshield.org (summary of over 1700 networks worldwide):
  
  - **Dshield nodes** are known IPs sending malicious or unwanted traffic during observation period.

We are grateful to Johannes Ullrich and the SANS Institute for making DShield logs available to us for the purposes of this study.
Network Representation

- **Key issue:** *Intrusion* as ‘entering a *community* to which one does not *belong*’ depends on network representation.

- **Standard bipartite graph:** $G_B=(V, U, E_B)$, where vertices $V$ are source IPs (srcIP), vertices $U$ are destination IPs (dstIP), and edges $E_B$ data flows from sources $V$ to destinations $U$. 
Network Representation

- Need for alternative network representation!
- One-mode projection: $G_p = (V, E_p)$ where two nodes are connected if and only if they share at least one common destination
- One-mode weighted projection: $G_w = (V, E_w)$, where also for each pair of nodes a weight (correlation coefficient) is assigned.
(Anti)social Behavior!

- **KEY FINDING:**
  - cliques in $G_p$ and $G_w$ capture the traffic common to *Internet communities*, while
  - *cut-vertices* (removal of which increases the number of components of these communities) are enriched with Dshield nodes;

- Hence, identification of *communities* and *cut-vertices* between them may be used to **capture antisocial behavior and detect source IPs sending intrusive traffic**.
Community Detection

- Standard modularity based community detection algorithm (Newman et. al, 2004) fails to distinguish DShield (known malicious) sources from defined ‘communities’:

<table>
<thead>
<tr>
<th>Size of Cluster</th>
<th># of Clusters</th>
<th># of DShields</th>
</tr>
</thead>
<tbody>
<tr>
<td>6784</td>
<td>1</td>
<td>158</td>
</tr>
<tr>
<td>986</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>&lt; 243</td>
<td>66</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>68</td>
<td>161</td>
</tr>
</tbody>
</table>
Cut-Vertices

**Method**
- Simple recursive search-and-prune strategy
- Greedy \( \frac{\text{max-flow}}{\text{min-cut}} \) algorithm

**Result**
- “Dshield source are highly enriched among cut-vertices”
- However, more robust concept of cut-vertices in social network literature is needed for detection
Detecting Malicious Sources

- Previous methods rely on high-degree intrusion attacks; detecting those is easy.

- **Our intrusion detection method is:**
  - aimed at identifying potential attackers that do not send a large number of probes in observation period (and so do not have high out-degree), and
  - based on thresholding of clustering and betweenness.
Detecting Malicious Sources

- A node is declared malicious if its *clustering (betweenness)* is below (above) threshold.
Detecting Malicious Sources

- A node is declared malicious if its clustering (betweenness) is below (above) threshold.
ROC graph (Receiver operating characteristic)

- Illustrate performance of a binary classifier system
- Systems have various discrimination threshold

| Predicted condition | True condition | Prevalence
|---------------------|----------------|----------------|
| Total population | Condition positive | Condition negative | $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$

| Predicted condition positive | True positive | False positive (Type I error) | Positive predictive value (PPV), Precision $\frac{\sum \text{True positive}}{\sum \text{Test outcome positive}}$
| Predicted condition negative | False negative (Type II error) | True negative | False discovery rate (FDR) $\frac{\sum \text{False positive}}{\sum \text{Test outcome positive}}$

| Accuracy (ACC) $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$ | True positive rate (TPR), Sensitivity, Recall $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$ | False positive rate (FPR), Fall-out $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$ | Positive likelihood ratio (LR+) $\frac{\text{TPR}}{\text{FPR}}$
| False negative rate (FNR), Miss rate $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$ | True negative rate (TNR), Specificity (SPC) $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$ | Negative likelihood ratio (LR-) $\frac{\text{FNR}}{\text{TNR}}$ | Diagnostic odds ratio (DOR) $\frac{\text{LR}_+}{\text{LR}_-}$


ROC graph

- Each curve are the a set of results.

- Curve lays in the upper-left is better than curve lays in lower-right.

- AUC – area under the curve
Results: Analysis of 1 Day Traffic

- **Example of 1 Day traffic:** for May 1, 2007 observed 29,585 sources sending 2,033,520 flows to 185,145 destination; 329 sources recorded in DShield logs *

- Area under the curve (AUC) is higher for clustering than betweenness for both $G_P$ (0.75 vs. 0.72) and $G_W$ (0.78 vs. 0.58).

* Note we calculate the clustering and betweenness values for Dshields not detected by a standard degree-based method.*
Results: Analysis of 14 Day Traffic

<table>
<thead>
<tr>
<th></th>
<th>Mean(AUC)</th>
<th>SE(AUS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clustering $G_p$</td>
<td>0.7440</td>
<td>0.0103</td>
</tr>
<tr>
<td>Betweenness $G_p$</td>
<td>0.7180</td>
<td>0.0084</td>
</tr>
<tr>
<td>Clustering $G_w$</td>
<td>0.7625</td>
<td>0.0080</td>
</tr>
<tr>
<td>Betweenness $G_w$</td>
<td>0.5621</td>
<td>0.0034</td>
</tr>
</tbody>
</table>

- AUC of each of four methods is stable over 14 Days with slight degrade over weekends.
- Topmost detection method is based on clustering on $G_w$, however
- In terms of intrusion detection, binary network representation $G_p$ provides competitive alternative to more complicated network $G_w$. 
Conclusion

Our detection system:

- Based on intuitive principle that network intrusion is antisocial communication;

- Detects malicious traffic generators without the need of signatures;

- Effective against low-degree (hard to detect) sources.