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

Per_{Exp}: expected percentage of suspicious hosts

SR_T: target sampling rate

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

Scalable spatial-temporal flow-correlation
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

<table>
<thead>
<tr>
<th>Intervals:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host A</td>
<td>homo-server</td>
<td>homo-server</td>
<td>homo-server</td>
<td>homo-server</td>
</tr>
<tr>
<td>Host B</td>
<td>similar-client</td>
<td>similar-client</td>
<td>similar-client</td>
<td>similar-client</td>
</tr>
</tbody>
</table>
Flow Capture: Synchronized IPs Detector

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

```
<table>
<thead>
<tr>
<th>Intervals:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host A</td>
<td>homo-server</td>
<td>homo-server</td>
<td>-</td>
<td>homo-server</td>
<td>homo-server</td>
<td>homo-server</td>
</tr>
<tr>
<td>Host B</td>
<td>non-homo-server</td>
<td>non-homo-server</td>
<td>non-homo-server</td>
<td>non-homo-server</td>
<td>non-homo-server</td>
<td>non-homo-server</td>
</tr>
</tbody>
</table>
```

```
Score

TH_Syn-Server

step_up = 1
step_down = 0.2
TH_{syn-server} = 4
```
Flow Capture: Sampling Probability Calculation

Sampling Sketch

- Prioritize Set of IP
- Sampling Probability
- Count

Syn IPs Detector

<Packet, Sampling_Prob> to Flow-Assembler

Target Sampling Rate

Counting Sketch

TCP
- SrcIP
- DstIP
- Cnt
- SYN
- SYN-ACK

UDP
- SrcIP
- DstIP
- Cnt
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

<table>
<thead>
<tr>
<th>Priority</th>
<th>Set of IPs</th>
<th>Sampling Rate</th>
<th>Cnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>syn-servers</td>
<td>p1</td>
<td>c1</td>
</tr>
<tr>
<td>2</td>
<td>syn-clients</td>
<td>p2</td>
<td>c2</td>
</tr>
<tr>
<td>3</td>
<td>Others</td>
<td>p3</td>
<td>c3</td>
</tr>
</tbody>
</table>

**Algorithm 2: Priority-based Sampling Algorithm**

```
begin
    budget = 1;
    foreach i = 1...n do
        if fi == 0 or budget ≤ 0
            pi = 0;
            continue;
        else
            pi = budget * fi;
            pi = pi > 1?1 : pi;
            budget = pi * fi;
            budget = budget – budget;
    return {p1, p2, ..., pn};
end
```

- \( P_t \): the pre-defined target sampling rate
- \( f_i \): the packet fraction for a priority out of all the packets
- \( \text{budget}_i \): the fraction of the sampled packets we would like to give to a particular priority
- \( \text{budget} \): available budget

\[
SR_{Actual} = \sum_{i=1}^{n} f_i * p_i = \sum_{i=1}^{n} \frac{f_i * \text{budget}_i * P_t}{f_i} = P_t \sum_{i=1}^{n} \text{budget}_i = P_t
\]
Flow Capture: Flow Assembler

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

\[
\text{Sampled Packets:} \\
p_1 \, p_2 \, \ldots \, p_n \hspace{1cm} b_1 \, b_2 \, \ldots \, b_n \\
\downarrow \hspace{3cm} \text{Flow Assembler} \\
(\text{SrcIP}, \text{SrcPort}, \text{DstIP}, \text{DstPort}, \text{Proto}) \\
\text{flow1:} \\ \text{flow2:} \\ \text{flow3:}
\]

- \(\text{Time}_{\text{start/end}}\): the start/end time of this flow
- \(\text{size}_{\text{actual}}\): flow size, \(\text{size}_{\text{Actual}} = n\)
- \(\text{byte}_{\text{actual}}\): the \# of bytes observed, \(\text{byte}_{\text{Actual}} = \sum_{i=1}^{n} b_i\)
- \(\text{size}_{\text{est}}\): the estimated flow size, \(\text{size}_{\text{Est}} = \sum_{i=1}^{n} \frac{1}{p_i}\)
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”
- $f_{ph_{\text{max}}}$: the maximum number of flows per hour
- $t_{\text{ime}_{\text{m}}}$: 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 ($M \leq N$), they are suspicious.

- Birch, a streaming clustering algorithm with good scalability
- Increase the “diameter” for discovering clusters to identify up to $\text{Per}_{\text{Exp}}$ hosts
Fine-Grained Detectors

- Fine-Grained detectors only focus on traffic of $\text{Per}_{\text{Exp}}$ 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

<table>
<thead>
<tr>
<th>Trace</th>
<th># of Pkts</th>
<th>Dur</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mar25</td>
<td>205,079,914</td>
<td>12h</td>
<td>header</td>
</tr>
<tr>
<td>Mar26</td>
<td>280,853,924</td>
<td>24h</td>
<td>header</td>
</tr>
<tr>
<td>Mar27</td>
<td>318,796,703</td>
<td>24h</td>
<td>header</td>
</tr>
<tr>
<td>Mar28</td>
<td>444,260,179</td>
<td>24h</td>
<td>header</td>
</tr>
<tr>
<td>Mar31</td>
<td>102,487,409</td>
<td>1.5h</td>
<td>full</td>
</tr>
</tbody>
</table>

**Table 1: Background Traces**

<table>
<thead>
<tr>
<th>Trace</th>
<th>Dur</th>
<th>Bots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bot-IRC-A</td>
<td>4days</td>
<td>3</td>
</tr>
<tr>
<td>Bot-IRC-B</td>
<td>4days</td>
<td>4</td>
</tr>
<tr>
<td>Bot-HTTP-A</td>
<td>4days</td>
<td>3</td>
</tr>
<tr>
<td>Bot-HTTP-B</td>
<td>4days</td>
<td>4</td>
</tr>
<tr>
<td>Bot-HTTP-C</td>
<td>4days</td>
<td>4</td>
</tr>
<tr>
<td>Bot-P2P-Storm</td>
<td>4days</td>
<td>2</td>
</tr>
<tr>
<td>Bot-P2P-Waledac</td>
<td>4days</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table 2: Botnet traces**

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

- 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”.
Evaluation

- Packet Sampling Algorithm

- The actual packet sampling rate is close to the target sampling rate
- High sampling rates for botnet C&C traffic
- Outperform another recent packet sampling algorithm (FlexSample)
Evaluation

- Cross-Epoch Correlation

<table>
<thead>
<tr>
<th>$SR_T$</th>
<th>0.01</th>
<th>0.02</th>
<th>0.03</th>
<th>0.04</th>
<th>0.05</th>
<th>0.06</th>
<th>0.07</th>
<th>0.08</th>
<th>0.09</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>48%, 0.1%</td>
<td>83%, 0.5%</td>
<td>96%, 1%</td>
<td>96%, 2%</td>
<td>100%, 3%</td>
<td>100%, 4%</td>
<td>100%, 5%</td>
<td>100%, 6%</td>
<td>100%, 6%</td>
<td>100%, 8%</td>
</tr>
<tr>
<td>0.025</td>
<td>52%, 0%</td>
<td>87%, 0.5%</td>
<td>100%, 1%</td>
<td>100%, 2%</td>
<td>100%, 3%</td>
<td>100%, 4%</td>
<td>100%, 5%</td>
<td>100%, 6%</td>
<td>100%, 7%</td>
<td>100%, 8%</td>
</tr>
<tr>
<td>0.05</td>
<td>48%, 0.1%</td>
<td>100%, 0.3%</td>
<td>100%, 1%</td>
<td>100%, 2%</td>
<td>100%, 3%</td>
<td>100%, 4%</td>
<td>100%, 5%</td>
<td>100%, 5%</td>
<td>100%, 7%</td>
<td>100%, 7%</td>
</tr>
<tr>
<td>0.075</td>
<td>48%, 0.2%</td>
<td>100%, 0.3%</td>
<td>100%, 1%</td>
<td>100%, 2%</td>
<td>100%, 3%</td>
<td>100%, 4%</td>
<td>100%, 5%</td>
<td>100%, 6%</td>
<td>100%, 7%</td>
<td>100%, 8%</td>
</tr>
<tr>
<td>0.1</td>
<td>39%, 0.3%</td>
<td>78%, 0.8%</td>
<td>100%, 1%</td>
<td>100%, 2%</td>
<td>100%, 3%</td>
<td>100%, 3%</td>
<td>100%, 5%</td>
<td>100%, 5%</td>
<td>100%, 7%</td>
<td>100%, 8%</td>
</tr>
<tr>
<td>1</td>
<td>30%, 0.5%</td>
<td>65%, 0.8%</td>
<td>96%, 1%</td>
<td>100%, 2%</td>
<td>100%, 3%</td>
<td>100%, 4%</td>
<td>100%, 5%</td>
<td>100%, 5%</td>
<td>100%, 7%</td>
<td>100%, 8%</td>
</tr>
</tbody>
</table>

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\textsubscript{T} (target sampling rate) and Per\textsubscript{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

<table>
<thead>
<tr>
<th>$SR_T$</th>
<th>0.01</th>
<th>0.02</th>
<th>0.03</th>
<th>0.04</th>
<th>0.05</th>
<th>0.06</th>
<th>0.07</th>
<th>0.08</th>
<th>0.09</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP(bots/23)</td>
<td>48%</td>
<td>83%</td>
<td>96%</td>
<td>96%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>FP(noises/1460)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Eliminate all the false positives
- Achieve high detection

With Flow-Corr ($Per_E = 5\%, M = 3$) with our approach

<table>
<thead>
<tr>
<th>$SR_T$</th>
<th>0.01</th>
<th>0.025</th>
<th>0.05</th>
<th>0.075</th>
<th>0.1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per of Pkts</td>
<td>1.7%</td>
<td>2.9%</td>
<td>2.1%</td>
<td>3%</td>
<td>4.3%</td>
<td>2%</td>
</tr>
<tr>
<td>Time</td>
<td>33s</td>
<td>39s</td>
<td>35s</td>
<td>40s</td>
<td>49s</td>
<td>33s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$SR_T$</th>
<th>0.01</th>
<th>0.025</th>
<th>0.05</th>
<th>0.075</th>
<th>0.1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Per_E = 5%$</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

- 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 $t = 0.0035 \times \text{"# of c-flows"}$, 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?