BotTalker: Generating Encrypted, Customizable C&C Traces

Han Zhang, Christos Papadopoulos
Outline

- Introduction
- Design and Implementation
- Use Cases
- Conclusions
Introduction

- Unencrypted C&C messages can be blocked by IDS using DPI techniques
- Encrypted C&C messages foil IDS
Problem - Network Operators

- How many bots does my IDS miss?

![Graphs showing unencrypted and encrypted botnet traces]
Problem - Research Community

- Research goal: develop detection algorithms to catch encrypted botnet traffic
- Need ground truth to evaluate detection algorithms
- Encrypted botnets make it hard to get ground truth
  - Is the captured botnet traffic encrypted?
  - Which part of traffic is encrypted?
  - Which encryption algorithm is used?
- Encryption makes it hard for researchers to develop new detection algorithms
Problem - Standards Community

- How do you develop IDS benchmarks?

- Need various types of encrypted botnet traffic, but ground truth is very hard to get

- Need customizable encrypted botnet traffic, but captured botnet traffic is static
Goal

- Use unencrypted botnet traffic with known ground truth to derive encrypted botnet traffic

- Develop a tool to generate realistic, customizable encrypted bot traffic with known ground truth
Outline

- Introduction
- Design and Implementation
- Use Cases
- Conclusions
Requirements

- Keep network level characteristics of botnet traces
  - timing information, flow duration, communication pattern

- Use of encryption algorithms used by botnets

- Emulate various encryption schemes

- Blend botnet traffic with background traffic
BotTalker Overview

Input

User-specified Parameters

Botnet Traces

Background Traces

BotTalker Converter

Output

Encrypted Botnet Traces

Encrypted Botnet & Background Trace

Unencrypted Botnet & Background Trace
# Control Knobs

<table>
<thead>
<tr>
<th>Knob</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption Algorithm</td>
<td>12 encryption algorithms (can be extended) e.g., XOR, AES</td>
</tr>
<tr>
<td>Encryption Scheme</td>
<td>Packet encryption, Flow encryption, SSL emulation</td>
</tr>
<tr>
<td>Which Flow to Encrypt</td>
<td>Single flow, Multiple flows, All flows</td>
</tr>
<tr>
<td>Which Protocol to Encrypt</td>
<td>Single protocol, Multiple protocols, All protocols</td>
</tr>
<tr>
<td>Traffic Blend Scheme</td>
<td>Introduce new hosts for bots, Assign bots to existing hosts</td>
</tr>
<tr>
<td>IP Replacement</td>
<td>Single IP, Whole Subnet</td>
</tr>
<tr>
<td>Timestamp Normalization</td>
<td>Normalize botnet trace timestamps to background trace</td>
</tr>
</tbody>
</table>
Use by Research Community

- Goal: test new detection algorithms

- Background Traces
- Unencrypted Botnet/Background Traces
- New IDS
- What is detection rate, TP, FP for new IDS? Does it work for encrypted botnet?
- BotTalker Converter
- Encrypted Botnet/Background Traces
- Unencrypted Botnet Traces
Use by Network Security Operators

- Goal: pick the right IDS for my network

Diagram:
- My Network
- Traffic Capturer
- Internet
- BotTalker Converter
- Botnet Traces
- Blended Traces
- IDS

Questions:
- Does it work well with my network? Maximum processed packets per second? Which IDS works for my network best?
Use by Standards Community

- Goal: develop IDS benchmarks
- Similar to DARPA 1999/2009 benchmarks
Outline

- Introduction
- Design and Implementation
- Use Cases
- Conclusions
Use Cases of BotTalker

- Use BotTalker to generate encrypted botnet traces
- Use traces to evaluate well known IDS’
  - BotHunter, Snort, Suricata
- Input:
  - 140 unencrypted kraken and butterfly botnet traces
  - Packet encryption using RC4
  - Flow encryption using RC4
  - SSL emulation
- Output:
  - Encrypted botnet trace fed into each IDS
Botnet Detection System - BotHunter

- BotHunter: a dialog-based botnet detection system
- Loosely ordered dialog for detection based on a score system
- How many bots missed by using encryption

<table>
<thead>
<tr>
<th>Traffic Encryption</th>
<th>Detected Bots</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>138/140</td>
</tr>
<tr>
<td>Yes</td>
<td>13/140</td>
</tr>
</tbody>
</table>

- Only 9.3% of encrypted bots are detected
IDS - Snort and Suricata

- **IDS**: Snort, Suricata
- **Rules**: Emerging threats open rule set

**Diagram**

```
Unencrypted Botnet Traces → Snort → Suricata → Triggered Alerts

Encrypted Botnet Traces → Suricata → Triggered Alerts
```

**Alerts Analyze and Compare**
Results: Snort

- **Input:** unencrypted botnet traces: 14982 alerts are triggered
- **Input:** encrypted botnet traces: 9926 alerts are triggered, 34% missed due to encryption
Results: Snort (cont.)

- Input: unencrypted botnet traces: 28 unique alerts: SMTP (1), DNS (9), Blacklist (1), DPI (17)
- Input: encrypted botnet traces: 11 unique alerts remain, 60.7% less
Results: Suricata

- **Input:** Unencrypted botnet traces: 13482 alerts, 37 unique alerts
- **Input:** Encrypted botnet traces: 8638 alerts (35.9% less), 16 unique alerts (56.8% less) are left

![Graph showing unencrypted and encrypted botnet traces comparison]
Outline

- Introduction
- Design and Implementation
- Use Cases
- Conclusions
Conclusions

- BotTalker is the first tool that can generate customized encrypted botnet traffic

- Using BotTalker we determined that encrypted botnet traffic evades BotHunter, Snort and Suricata

- Provided tool to help research, network operator and standards communities
Questions

If you are interested, contact me: zhang@cs.colostate.edu
The tool will be released soon