Detection of malicious network traffic using behavior signatures

Christian J. Dietrich

2010-07-07

SPRING 5, Bonn

Contents

1. Problem statement and requirements
2. Discussion of approaches
3. Outlook

1. Problem statement and requirements

Problem statement (1/2)

- Malicious remote controlled software, i.e. bots, cause lots of problems in today‘s Internet
- Malicious network traffic examples
  - Related damage (spam, infections, DDoS, credential theft, click fraud...)
  - Command and control (C&C) network traffic
  - Deluding network traffic
- **None of these need to be obvious**
  - e.g. a trojan exfiltrates data encrypted and steganographically hidden in an image uploaded to flickr / video to youtube / blog comment...

Problem statement (2/2)

- Most existing botnet detection methods **require** related damage/attack traffic! (BotHunter, BotMiner)
- Existing detection is based on static criteria that are *supposed* to be characteristic for botnets, e.g.
  - Specific payload byte signatures (Botzilla, TAMD)
  - Regularity/periodicity of network behavior (BotSniffer, BotMiner, Intel Canary)
  - Destination access patterns (TAMD)
- C&C is more and more **encrypted**, thus payload byte signatures are no longer applicable
- Is it possible to detect malicious remote controlled software based on its **network behavior**?

Requirements of a behavior-based bot detection method

- Must-have
  1. Behavior signatures should be dynamic and adapted as necessary (automated signature extraction)
  2. Behavior signatures should be as independent as possible from the learning environment
  3. Detection should be adequately accurate, i.e. should have very low misclassification rates
- Nice to have
  4. Not solely depend on attack/damage traffic
     - Detection should be based on any kind of traffic that is present even if no attack takes place, such as C&C traffic
     - Attack traffic may support detection

2. Discussion of approaches

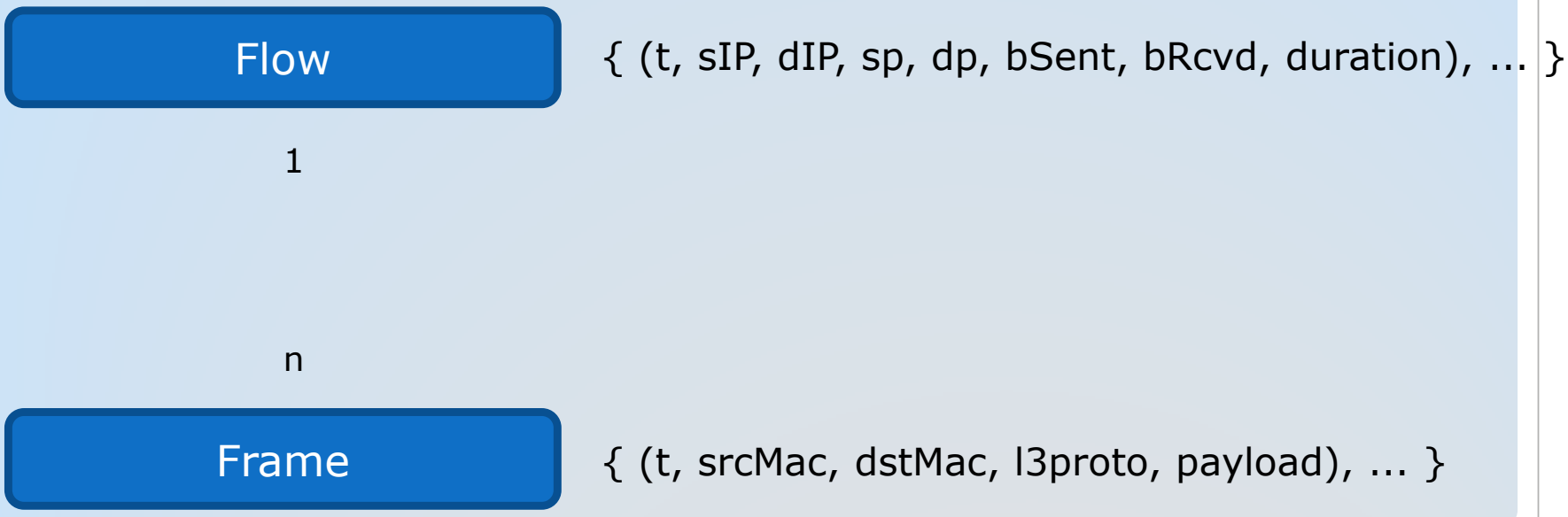
Approach:
Flow Based Botnet Detection

- Extract features from bot network traffic samples
  - A) in a contained environment (sandnet)
  - B) in the wild (using A/V as sensors)
- Extract features from legitimate network traffic
- Label the feature sets (malicious/benign)
- Aggregate feature sets
  - Detect infected hosts
    => aggregate by source IP address
- Build a model (machine learning, especially SVM)
- Apply the classifier to features extracted from live network traffic at network egress points

Challenges

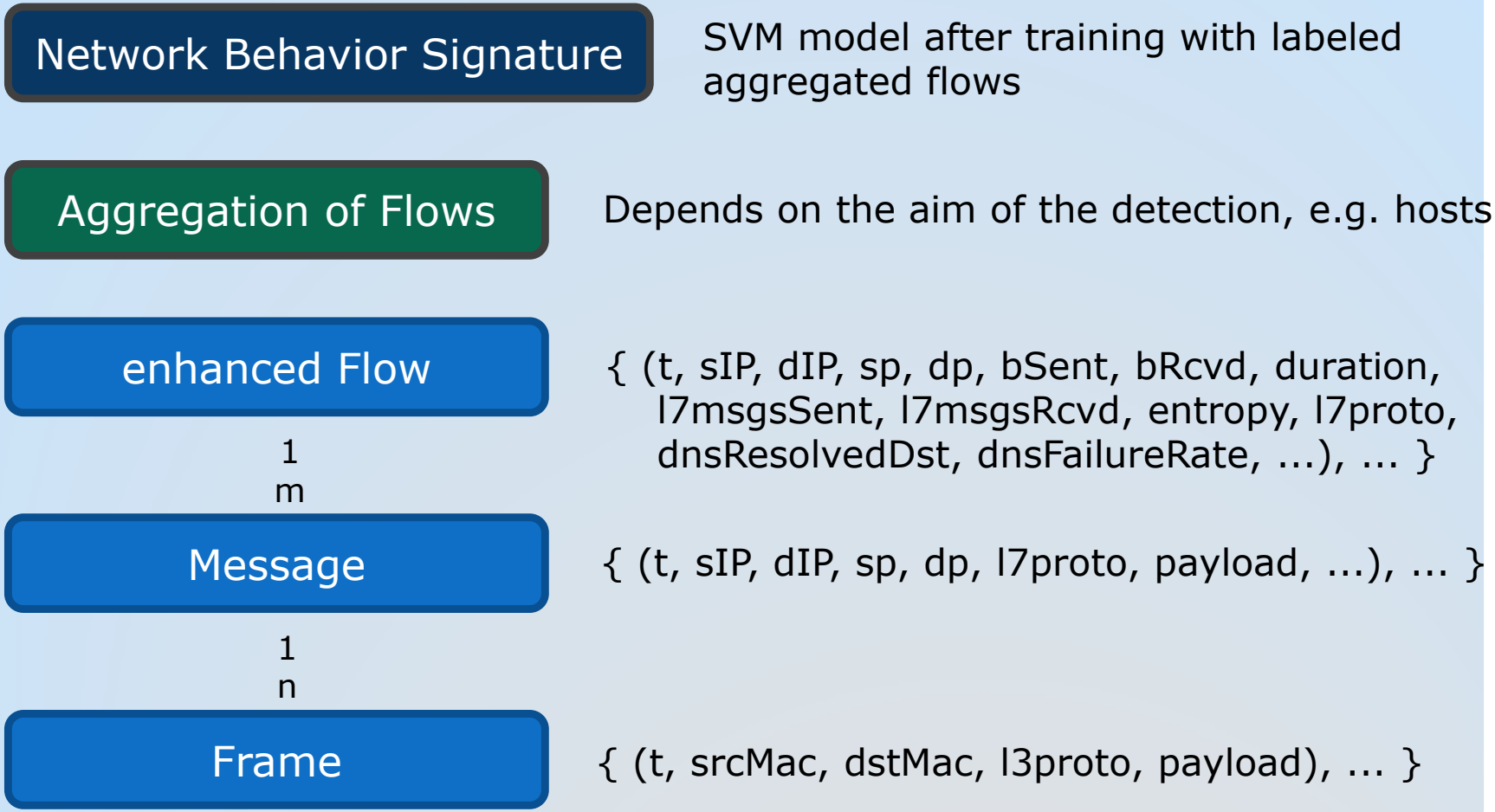
- Which features shall be extracted (abstraction)?
  - Network **flow-level features** (duration, src&dst, ports, l4proto, l7msgs, bytes sent/rcvd, entropy, dst_domain, l7proto, ...)
- Formal definition of a behavior signature?
  - Aggregation of flows as a set (no order)
  - Express as a sequence of flows (causality)
  - SVM model based on training on an aggregation of flows
- Unclean training data
  - Clean network traffic is difficult to acquire
  - Bot traffic may contain legitimate-looking flows (e.g. a Google search)
  - Requires a **robust learning method**
- Lots of related work

Classic NetFlow (sFlow similar)



t=timestamp, sp=src port, dp=dst port, bSent=bytes sent, bRcvd=bytes received

From frames to flows to NBS



t=timestamp, sp=src port, dp=dst port, bSent=bytes sent, bRcvd=bytes received

Aggregation of Flows

- Aggregate a set of flows
  - Order of the flows is not important
- Aggregate a sequence of flows
  - Order of the flows is important
  - Implies causality of flows
- Aggregation criteria
  - Aggregate by source host / IP address
  - Aggregate by execution of a bot binary
  - Sample sets of flows over time

Network Behavior Signature

- „Compare the resulting aggregations“
- Clustering of resulting aggregations
  - Are there „similar“ aggregations among different bots?
- Aim: learning an SVM model based on the flow aggregations

- Define a network behavior signature as the model that results from SVM-based learning

Building subsets of network traffic for training and detection

- Considering all network traffic might result in performance problems
- Are there reasonable subsets of network traffic that suffice for training and detection?
  - Certain layer 7 protocols, e.g. HTTP
  - Sampling of network traffic

Restrict to HTTP traffic

- HTTP is used more often by bots, especially as underlying C&C protocol
- Again flow based detection, but restrict to HTTP network traffic
  - Many false positive candidates
  - Thus, try to even restrict to HTTP C&C traffic
- Challenges
  - Definition/Identification of C&C traffic
  - Evaluation nearly impossible

Interim Findings

- Botnet C&C network traffic is not detectable in mixed network traffic!
  - Botnet C&C is effectively a covert channel
  - There are always means to hide C&C communication in today's Internet traffic
  - Separation of C&C network traffic and non-C&C traffic is impossible, especially given an initial abstraction layer such as network flows

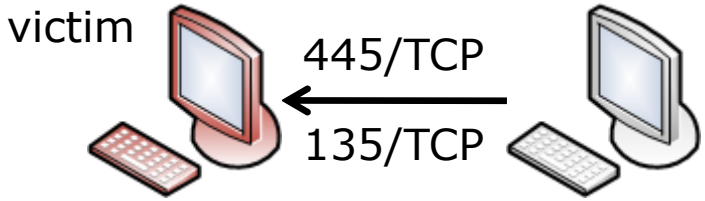
1. Restricting the network traffic: risk of losing what is important
2. Bot detection based on the presence of network traffic (no matter what kind of traffic this is)

➔ **Hence look at further attack/damage functions**

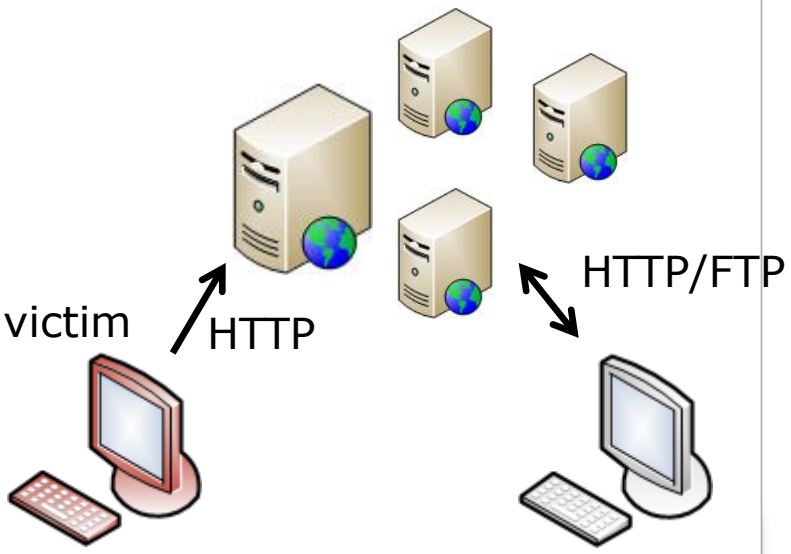
3. Outlook

Indirect infections

- Focus on infections
  - So far: **direct** infections (mostly via 445 or 135/TCP)
  - Targets server/daemon software



  - Today: **indirect** infections, i.e. no direct communication between infector and victim (e.g. drive-by infections)
  - Targets client software



Thanks for your attention.

Questions?