HMM Profiles for Network Traffic Classification

Charles Wright, Fabian Monrose and Gerald Masson

Johns Hopkins University
Information Security Institute
Baltimore, MD 21218

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Models
- Experiments
- Results
- Future Directions

Problem Description

- Given a packet trace, what protocol(s) generated it?
  - using only time & size
- Why:
  - backdoors
  - bad users
    - everything runs on port 80
    - SSH tunnels
  - identify new, unknown traffic

The Power of Size and Timing Data

- Identifying stepping-stone hosts
  - Zhang and Paxson, USENIX Security 2000
- Identifying web pages in encrypted traffic
  - Q. Sun, et al., Oakland 2002
- Recovering passwords from SSH sessions
  - D. Song, et al., USENIX Security 2001

Can We Do More?

- Existing work is application-specific
  - Let's try to move down one layer in the protocol stack
- Existing work consists mostly of attacks
  - Can we use this defensively?

Immediate Goals

- Modeling
  - Construct packet-level models for common application protocols, using very little per-packet data.
  - This is new
- Classification
  - Given a TCP connection, what protocol generated it?

Other Applications ?

- Profiling application processes using system calls
  - data has the necessary structure
- Profiling users with shell commands
  - ???

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Models
- Experiments
- Results
- Future Directions

Background: Intrusion Detection

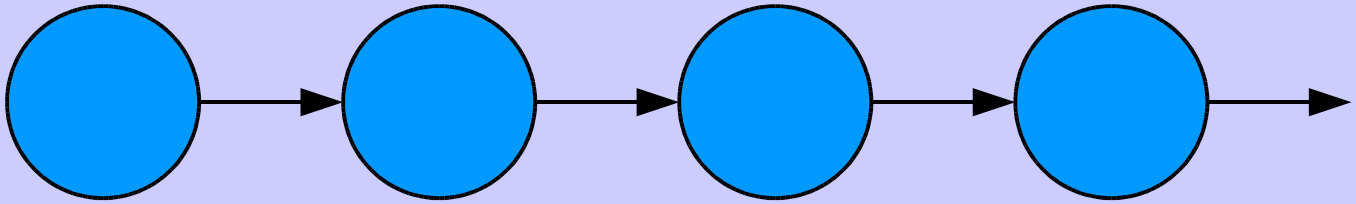
- Biology:
  - S. Forrest, *et al.*, Oakland 1996, CACM '97
- Sequence Alignment:
  - S. Coull, *et al.*, ACSAC 03
- Decision Trees:
  - Early, *et al.*, ACSAC 03
  - classifier only, no model
- HMM methods: Ye, Lane, DuMouchel
  - fully-connected, no use of structure

Background: HMM Profiles

- Described by Krogh et al. and Eddy et al. in the bioinformatics literature, for multiple sequence alignment.
- Compares favorably to other methods
- Great when you have lots of data
  - Doesn't require all pairwise alignments
- Good for regular data with frequent small irregularities
  - Lots of Inserts and Deletes

HMM Profiles

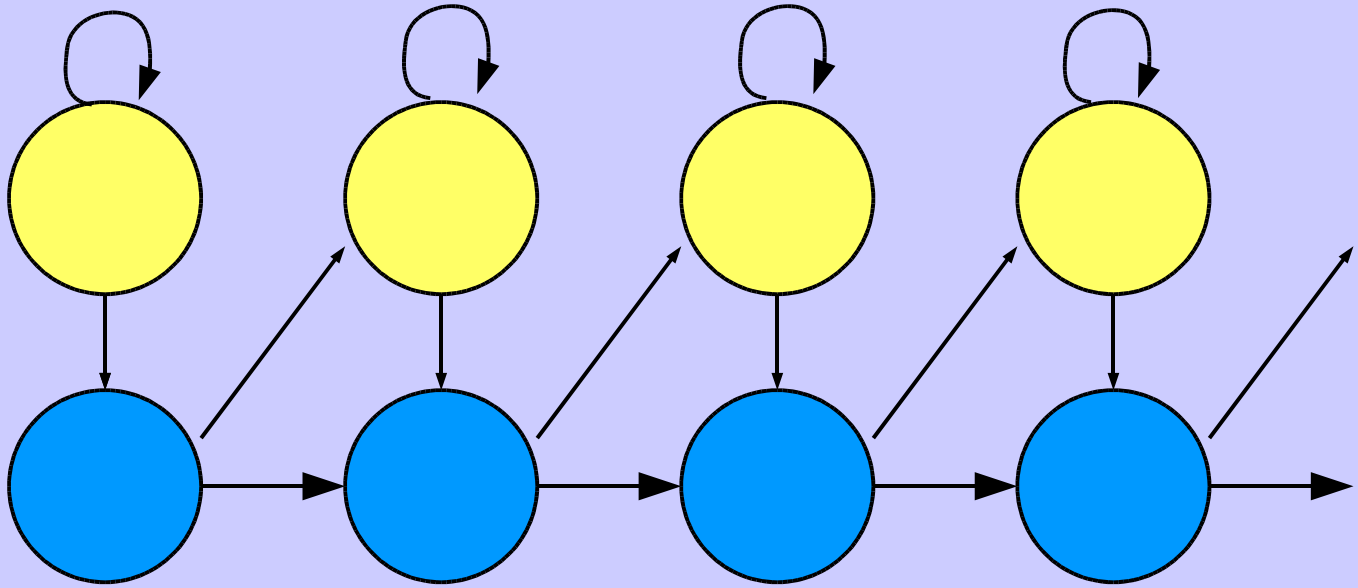
- Left-right Hidden Markov Models built around a central chain of “Match” states



- These states capture the essential behavior we're interested in

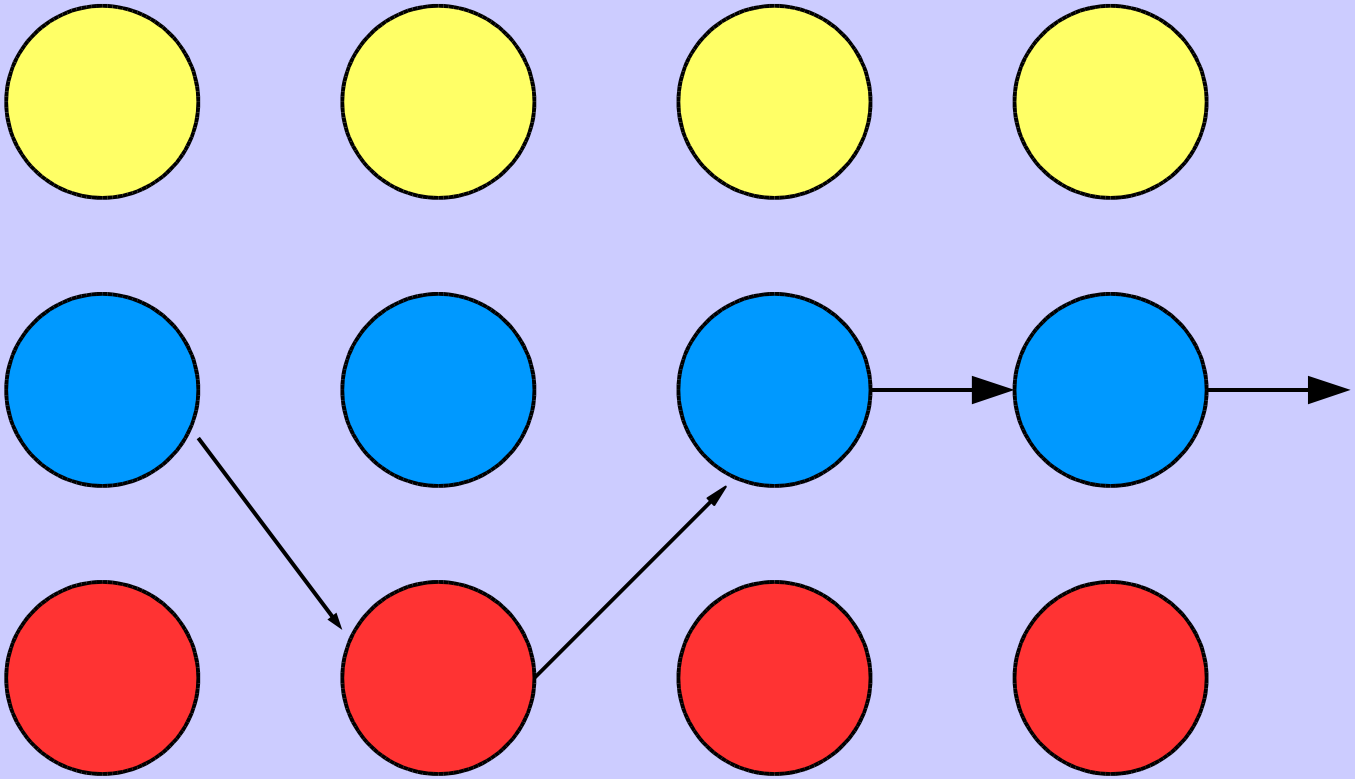
HMM Profiles

- “Insert” states represent unusual data in an otherwise usual sequence



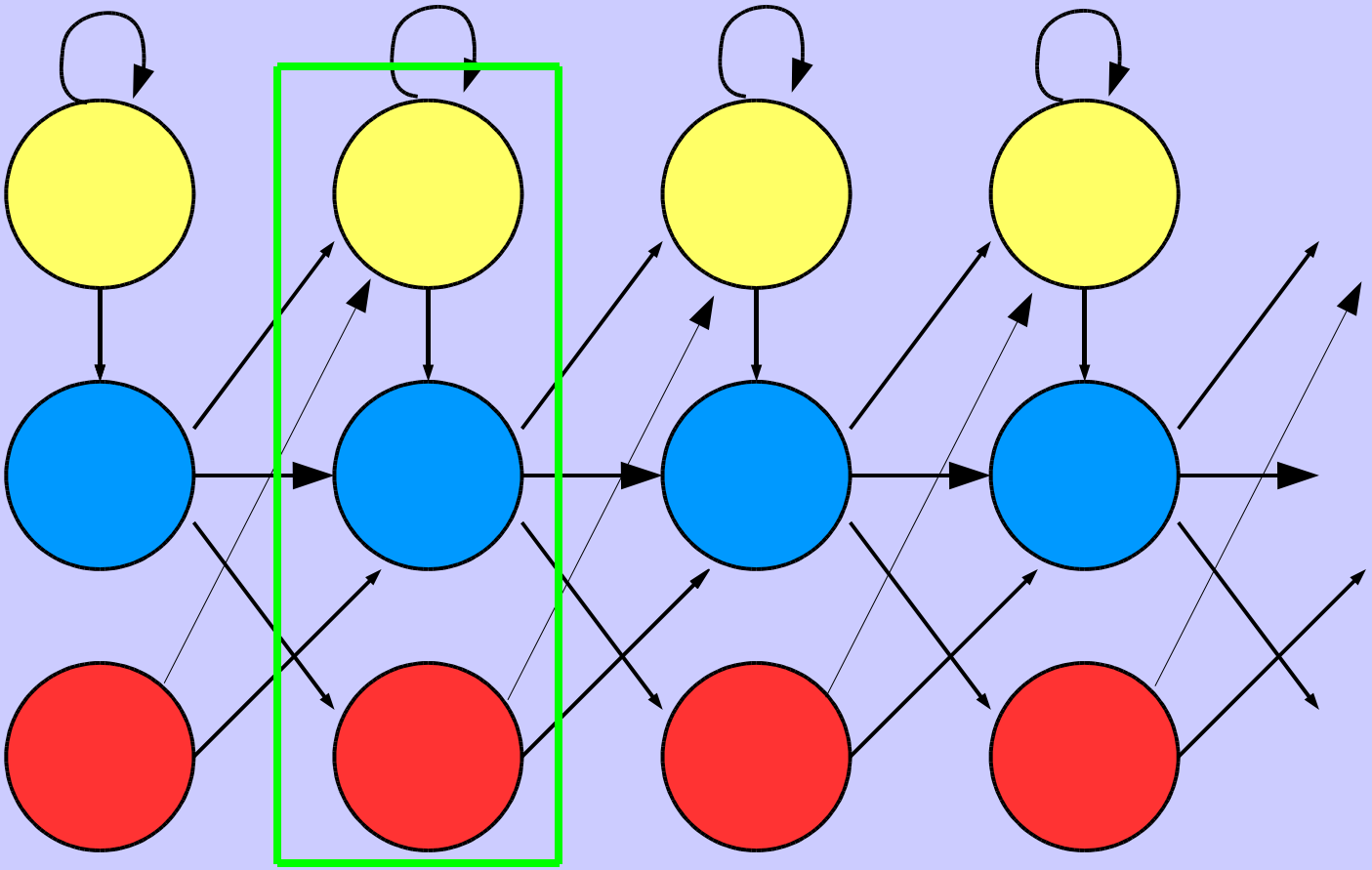
HMM Profiles

- “Delete” states allow some usual data to be omitted

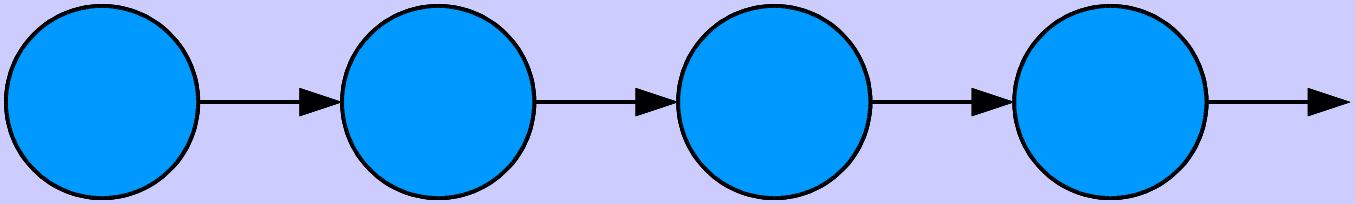


HMM Profiles

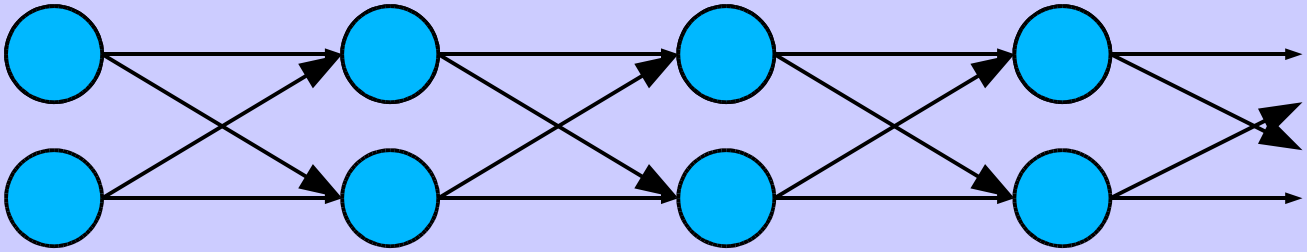
- Visually, we have a column of states for each position in the chain



Modification: Two Match States



- Replace each Match state with two states:



  - One for packets from the Client
  - One for those from the Server

Why Two Match States?

- Intuition
  - interactivity: lots of cross-talk
  - data transfer: most packets go the same direction
- Measurements back this up
  - ACF for lag 1 shows correlation between successive packets
- Our choice represents a compromise between parsimony and precision

Interpreting the Models

- *Match* States
  - Represent typical packets from Client to Server, or Server to Client
- *Insert* States
  - Represent duplicate packets and retransmissions
- *Delete* States
  - Represent packets lost in the network, or dropped by the sensor

A Caveat

- This technique relies on predictable structure and patterns in the protocol traces
- Most application protocols have such structure
  - it's defined in their RFC's
- Some interactive protocols don't
  - For example: Telnet, SSH
  - This is a problem

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Modeling
- Experiments
- Results
- Future Directions

Modeling

- We build two models for network sessions, using only a single feature in each:
  - Packet size
  - Packet interarrival time
- These features are still intact after encryption

Special Considerations: Packet Sizes

- Problem:
  - We only have unencrypted data
  - Some modes of encryption (block ciphers) will mask the packet's true size
- Solution: binning
  - We round the cleartext packet sizes up to the next whole multiple of the block size.

Special Considerations: Packet Interarrival Times

- Problem: Interarrival times take on a wide range of values
  - And their distribution is heavy-tailed
- Problem: The network creates jitter in interarrival times

Special Considerations: Interarrival Times

- So it's harder to model this data
  - discrete models need a huge alphabet
  - discrete models could be confused by jitter
- Continuous models might be more appropriate
  - But they're harder to work with

Special Considerations: Interarrival Times

- Solution: use log(time) instead of time
  - Now we can bin the data and build a discrete-density HMM
  - This also masks some of the jitter
  - Use different sample rates to get larger or smaller bins

Building Models

- Initial model
  - Manually constructed, “hard-coded”
  - number of columns = average length of the sessions in the training set
  - all packets are equally likely, in all positions in the session

Building Models

- Train the initial model on a sample of the traces
  - Re-estimate HMM parameters using the Baum-Welch algorithm (EM)
  - Typically 2 or 3 iterations are enough

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Modeling
- Experiments
- Results
- Future Directions

Classification

- Many ways to choose a model for a sequence:
  - Maximize log p(sequence | model)
    - used in the paper
  - Maximize log p(Viterbi path)
    - provides a better alternative
    - results not discussed here
  - Lots of other possibilities

Experimental Setup

- tcpdump data from MITLL Intrusion Detection Evaluation
  - a week of traces
  - 1-2 thousand examples of most protocols
- Data from the Internet link at GMU
  - 10 hours of traces
  - tens of thousands of examples for most

Experiments

- First, we try to classify just a few protocols
  - FTP, Telnet, HTTP, SMTP
- Comparison with earlier work
  - Early, *et al.*
- There's an error in the paper:
  - Early, *et al.* uses 5 protocols, not 4
  - apologies to the authors

Experiments

- Examine a more realistic mix of 9 protocols:
  - non-interactive traffic:
    HTTP, HTTPS, SMTP (2 directions), FTP (control and data)
  - interactive traffic:
    Telnet, SSH, AOL Instant Messenger
  - Initially, some appear difficult to characterize:
    SSH, FTP control, AIM

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Modeling
- Experiments
- Results
- Future Directions

MITLL: Size Model Results

Protocol	Classifications (percent)				
	ftp	smtp	telnet	http	none
ftp	99.4	0.0	0.0	0.0	0.6
smtp	0.1	97.9	1.0	0.0	1.0
telnet	3.1	0.0	21.7	1.6	73.6
http	0.0	0.1	0.5	96.9	2.4

Table 1: Confusion Matrix for the Size-Based Classifier on MITLL traces (blocksize = 64 bytes)

MITLL: Timing Model Results

Protocols	Classifications (percent)				
	ftp	smtp	telnet	http	none
ftp	82.2	0.0	0.0	0.0	17.8
smtp	2.5	96.6	0.0	0.2	0.7
telnet	1.3	2.6	14.2	0.0	81.8
http	0.3	0.8	0.3	95.3	3.3

Table 2: Confusion Matrix for the Timing-Based Classifier on MITLL traces (sampling rate = 1)

MITLL Results: discussion

- Within 5% of Early, *et al.* results
  - except for Telnet
  - difficulty with different protocols
- However, this dataset is not realistic
  - based on LAN traffic
  - low data rate
  - small number of hosts involved

GMU: Size Model Results

	Classification Probability (percent)									
Protocol	aim	smtp-out	smtp-in	http	https	ftp-data	ftp	ssh	telnet	none
aim	81.6	1.9	0.9	0.4	2.5	0.5	0.5	3.4	2.7	5.6
smtp-out	4.0	65.6	12.7	0.0	0.6	0.3	7.2	0.8	4.4	4.4
smtp-in	1.1	11.4	70.2	0.0	0.3	0.2	2.0	0.6	11.9	2.3
http	0.2	0.1	0.1	81.1	6.2	7.5	0.1	0.6	0.8	3.6
https	0.5	4.1	0.2	2.8	76.5	1.9	0.1	2.2	3.2	8.5
ftp-data	2.1	4.4	5.1	12.1	4.0	62.7	0.2	0.4	5.9	3.1
ftp	1.0	22.7	2.9	0.1	0.0	2.0	62.7	3.9	4.7	0.0
ssh	7.0	1.4	0.7	1.1	14.7	0.4	2.5	42.1	1.9	28.1
telnet	2.6	4.1	1.0	4.4	8.0	4.7	2.4	5.4	42.0	25.5

Table 4: Confusion Matrix for the Size-Based Classifier on GMU Traces (blocksize = 32 bytes)

GMU: Timing Model Results

	Classification Probability (percent)									
Protocol	aim	smtp-out	smtp-in	http	https	ftp-data	ftp	ssh	telnet	none
aim	86.4	0.4	0.6	0.5	1.7	0.1	1.7	3.9	1.8	2.7
smtp-out	2.3	66.8	9.4	1.6	2.3	0.4	2.9	4.3	4.8	5.0
smtp-in	2.3	10.2	67.1	2.0	3.7	1.5	3.3	6.2	1.6	2.0
http	1.3	3.0	1.9	67.6	10.9	7.0	2.2	2.8	0.7	2.4
https	1.1	3.3	2.9	12.4	56.0	4.9	2.8	7.6	2.1	6.9
ftp-data	0.0	2.4	5.5	13.7	6.9	61.6	1.1	2.8	1.4	4.5
ftp	4.0	3.9	5.9	0.8	2.9	5.5	49.8	13.3	9.2	4.9
ssh	10.6	1.8	3.8	1.6	5.2	1.9	4.1	39.0	9.7	32.4
telnet	6.2	2.8	2.1	0.6	3.2	0.8	6.5	13.0	31.7	33.1

Table 5: Confusion Matrix for the Timing-Based Classifier on GMU Traces (sampling rate = 3)

GMU Results: discussion

- AIM is less interactive than we'd thought
- FTP control is interesting
  - judging by size, it's confused with SMTP
  - judging by timing, it's confused with SSH
- Increasing precision doesn't always improve overall performance
  - But it does tend to reduce the most frequent errors

Why Size Matters

- Most packet sizes are not dependent on ever-changing network conditions
  - we will see *some* duplicates, retransmissions, and omissions
- Interarrival times depend on the state at every router in the packets' paths
  - this is true for *every* pair of packets in the sequence

Overview

- Problem Description
- Background
  - Intrusion Detection
  - HMM Profiles
- Modeling
- Experiments
- Results
- Future Directions

Simple Improvements

- Modeling
  - model surgery
  - simulated annealing
- Model Selection
- Better Classifier(s)

Improving the Classifier

- Use the Viterbi Path's probability
  - already implemented
  - no real drawbacks
- Looking at other schemes
  - average transition or emission probabilities
  - information gain:

$$IG = H(model) - H(model \mid sequence)$$

Future Directions

- Unsupervised Learning: Model-Based Clustering
  - will be useful for analyzing the SSH data
- Two-variable Models
  - use both size AND time
  - work in progress now
  - The Curse of Dimensionality

Conclusions

- Validates work by Early, *et al.* and Coull, *et al.*
- Demonstrates the necessity of testing on real data
- The HMM profile technique shows promise
  - Although naïve in implementation, we still achieve results comparable to prior work
  - Can classify wide-area traffic
- There are areas for improvement