

# APE: Adaptive Prevention Environments

## A Framework for Adaptive Virus Detection and Infection Prevention

Steve Martin, Blaine Nelson, Anil Sewani

CS262A: Advanced Topics in Computer Systems, University of California at Berkeley

### The Problem

- Viruses are a huge global problem.
  - Nimda: \$635 million in cleanup
  - Code Red: \$2.62 billion
  - Love Bug: \$8.75 billion
- Detecting a virus is easy with signature matching methods.
  - McAfee, Symantec, etc
- However, generating this signature requires human action.
  - And also humans to install the update!
  - Viruses can spread so quickly that it is no longer possible for humans to react fast enough.
- Therefore, need a first line of defense that slow or stop the infection rate of new viruses.
  - Current solutions not adaptive enough!

### Network Component

- Built threaded packet sniffing and multiqueue components.
  - Uses Linux pcap and pthreads libraries.
  - Will be used in the future to expand work.
- For current prototype, use captured email from a variety of sources.
- Built an email traffic simulator to simulate a busy network.
  - Builds a simulated capture trace using email from several people.
  - Highly customizable to give different network traffic views.
- Can arbitrarily 'infect' a given user in the trace with a virus at any time.
  - Can change all aspects of infected user's email activity.
- Email traffic is modified in the trace to match research observations on infected machine activity.

### Learning Features

- Given a single email or series of emails, how to come up with values that describe them?
  - What features best determine virus emails from non-virus emails?
- Our current feature set:
  - Per-email features:
    - Number of attachments
    - Total size of attachments
    - Number of characters in the subject text
    - Number of words in body
  - Per-flow features (measured over a sliding window of emails):
    - Frequency of email sending
    - Ratio of emails with attachments to emails without attachments
    - Number of different address sent to
- Feature vector generator written in Perl for ease of rapid modification to add/modify features.

### Our Approach

- Use unsupervised machine learning to monitor network behavior.
  - Our technique uses Gaussian Mixture Models.
- Instead of looking for specific viruses, learn a model of network traffic and use it to classify abnormalities.
  - Leverage the fact that an infected machine has a different network activity signature.
- Model changes over time to adapt to network usage patterns.
  - Use architecture similar to that of the Click router to handle high load and for customizability.
- Initially, limit scope to email viruses.
  - Method could be easily expanded to the packet level later.

### Model Component: Theory

#### EM Algorithm

- E Step

$$\tau_n^{k(t)} = p(Z^{k(t)} = 1 | x_n, \theta^{(t)})$$

- M Step

$$M_k^{(t)}(b) = \sum_{n=1}^N \tau_n^{k(t)} (x_n)^b$$

$$\pi_k^{(t+1)} = \frac{M_k^{(t)}(0)}{N} \quad \mu_k^{(t+1)} = \frac{M_k^{(t)}(1)}{M_k^{(t)}(0)}$$

$$\Sigma_k^{(t+1)} = \frac{M_k^{(t)}(2)}{M_k^{(t)}(0)} - \left( \mu_k^{(t+1)} \right)^T \left[ \mu_k^{(t+1)} \right]$$

#### Extreme Value Stats

- Gumbel Distribution

$$P_{extreme}(x | m) = e^{-e^{-\sigma_m^{-1}(x-\mu_m)}}$$

$$\mu_m = \sigma_m^{-1} + 1/2 \cdot \sigma_m (\ln \ln m + \ln 2\pi)$$

$$\sigma_m = (2 \ln m)^{-1/2}$$

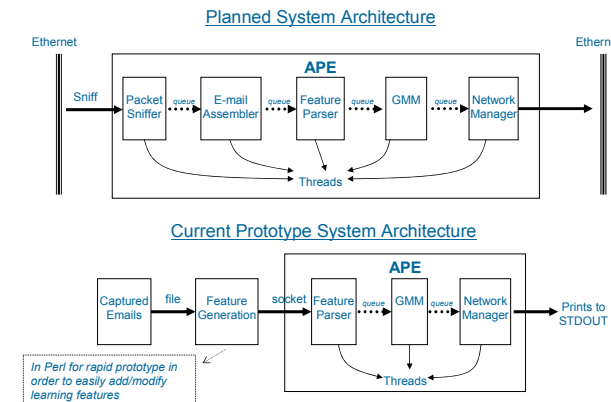
- GMM Extrema

$$P(\mathbf{x} | m) = P_{extreme}(h_{k^*}(\mathbf{x}) | m \cdot \pi_{k^*}^{(t)})$$

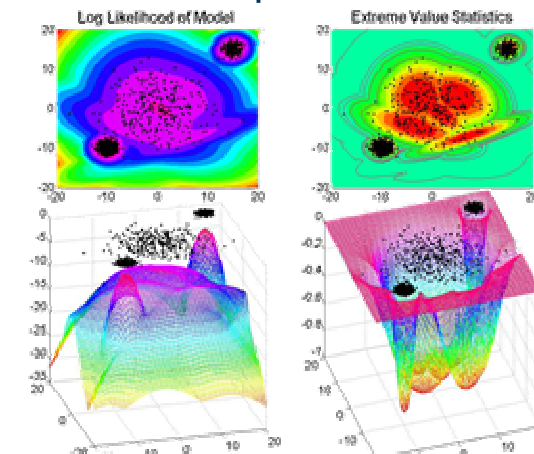
$$k^* = \operatorname{argmin}_k (h_k(\mathbf{x}))$$

$$h_k^{(t)}(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu_k^{(t)})^T (\Sigma_k^{(t)})^{-1} (\mathbf{x} - \mu_k^{(t)})}$$

### System Overview



### Model Component: Results



### Preliminary Results

- Tested four runs on set of ~300 emails each, some of which were simulated to be from one of 3 viruses.
  - Two popular email viruses with known behavior parameters, and one never-seen-before virus we created to have different behavior.
- Used a subset of our features to avoid over fitting.
  - Still working to refine complete feature set to more fully span 7-dimensional space.

Table 1. APE Performance results

Virus Name	Total Emails	# Infected Emails	False Positives	False Negatives	% Correctly Classified
None	208	0	19	N/A	90.86
AnnaKourmikova	258	50	12	14	89.92
W32.Sobig.F	308	100	5	60	78.90
Research Virus	255	50	11	13	90.58

### Future Work

- Test with actual network with real viruses
  - We have the infrastructure to capture live emails.
  - Would like to work with real network and see if we can detect infections.
- Expand and refine features based on virus research
  - Could improve results drastically
- Move detection to packet level
  - Email viruses are the tip of the iceberg!
- Refine model, evaluate other machine learning techniques.
  - Add splitting to GMMs to enhance fit, build better classification techniques, etc.
- Move towards planned architecture as components solidify
  - Right now, some components outside ape for ease of doing research.