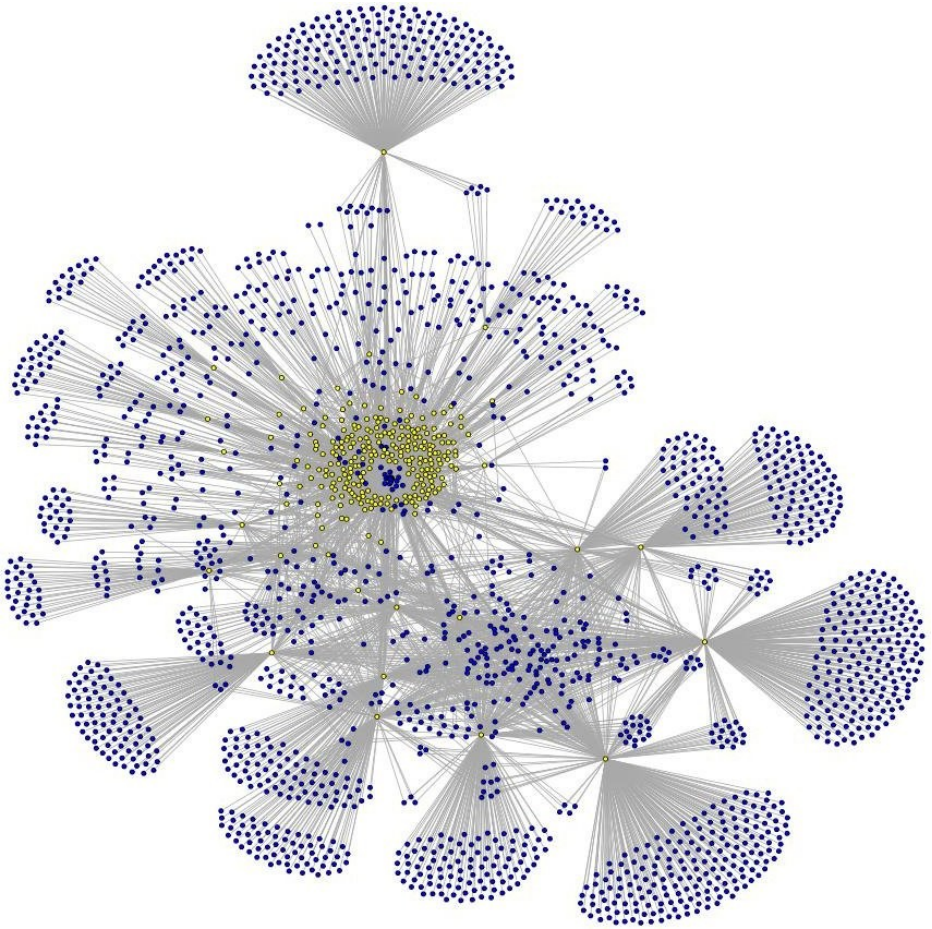
Realtime Change Detection & Automatic Network Response

Alex Brugh (presenter)
Mike Fisk, Josh Neil, Paul Ferrell, Scott Miller, Danny Quist
Los Alamos National Laboratory
Advanced Computing Solutions
LA-UR 09-08132

Outline

- Use of Flow in Change Detection
  - Current Methods
  - Areas of research
- Response
  - Automated framework
  - Methods



Change Detection

- We'd like to detect misuse of the network
- Most researchers have attempted to train systems to detect malicious traffic
  - But there is no realistic, representative training data
  - Over-fitting the training data is always a risk
- Our approach: focus on detecting change in the status quo
  - Change can be detected without training data
  - Statistically significant changes tend to be interesting to security and/or operations
  - Finding only future attacks isn't perfect but isn't bad

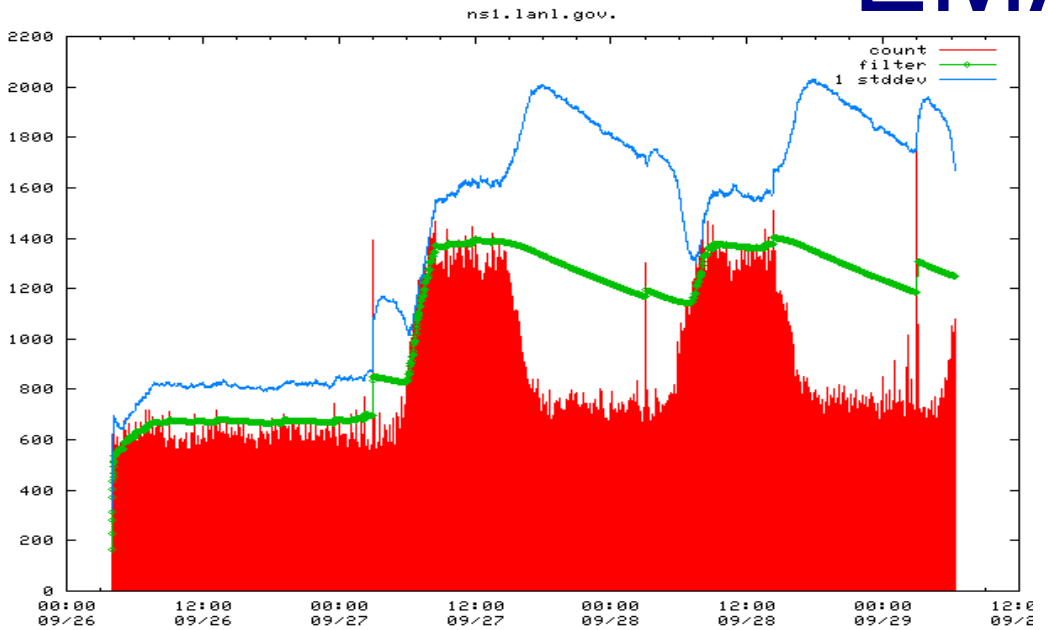
Flows

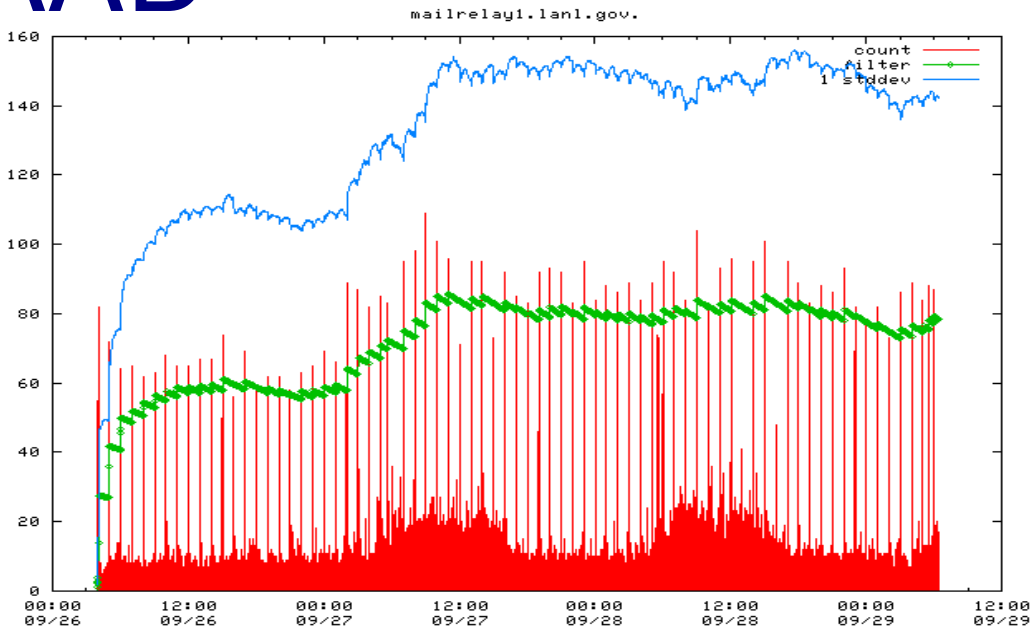
- Flows are binned by time
  - e.g., X counts per minute
- Many possible features
  - Connections from 1 host to many, from many to 1
  - Protocol distribution (e.g, TCP/UDP)
  - Ports
  - Bytes or packets sent/received
  - Graphs

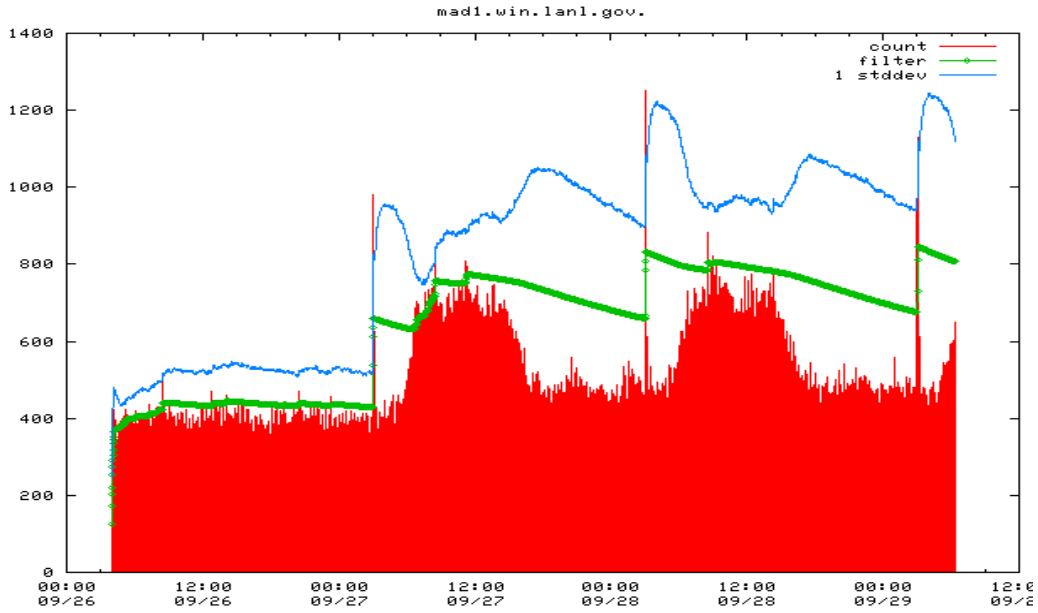
EMAAD

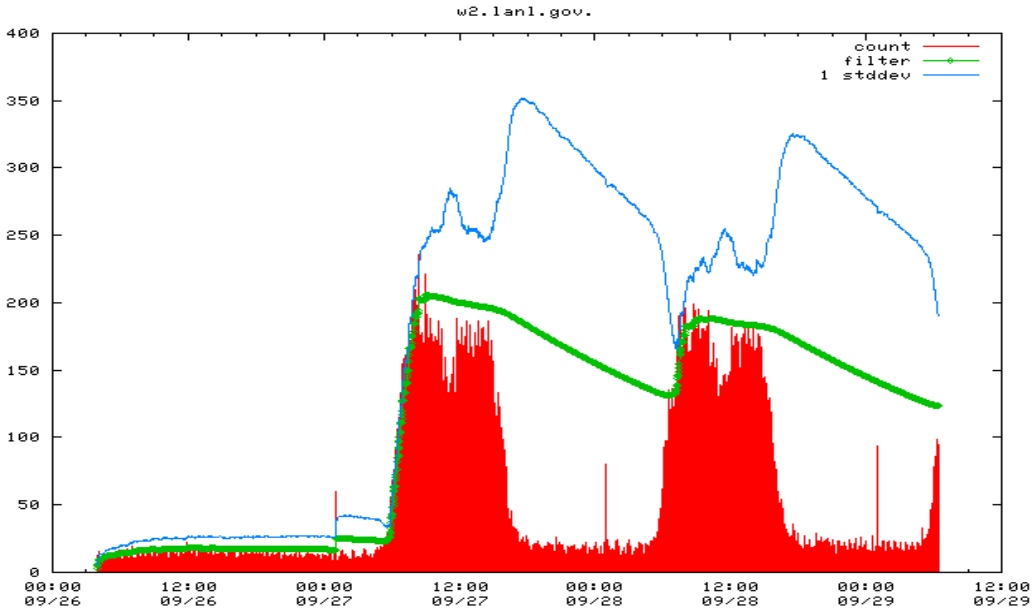
- Exponential Moving Average Anomaly Detector
- Models number of unique connections originating from a given host per unit time
  - IP to IP, ignores ports
- How it works
  - Assumes Gaussian model
  - Asymmetric moving average
    - Adapt quickly to increases
    - Decay slowly with decreases
  - Model built for each host (little state)

EMAAD


ns1.lanl.gov.
count
filter
1 stddev


mailrelay1.lanl.gov.
count
filter
1 stddev


mad1.win.lanl.gov.
count
filter
1 stddev


w2.lanl.gov.
count
filter
1 stddev

EMAAD

- Sub-minute response times
  - Continually recalculates as a bin updates
  - Handles 10s of thousands without
- Alarm causes
  - Host discovery (nmap, ping sweeps, etc)
  - Misconfiguration
  - HP printer administration software

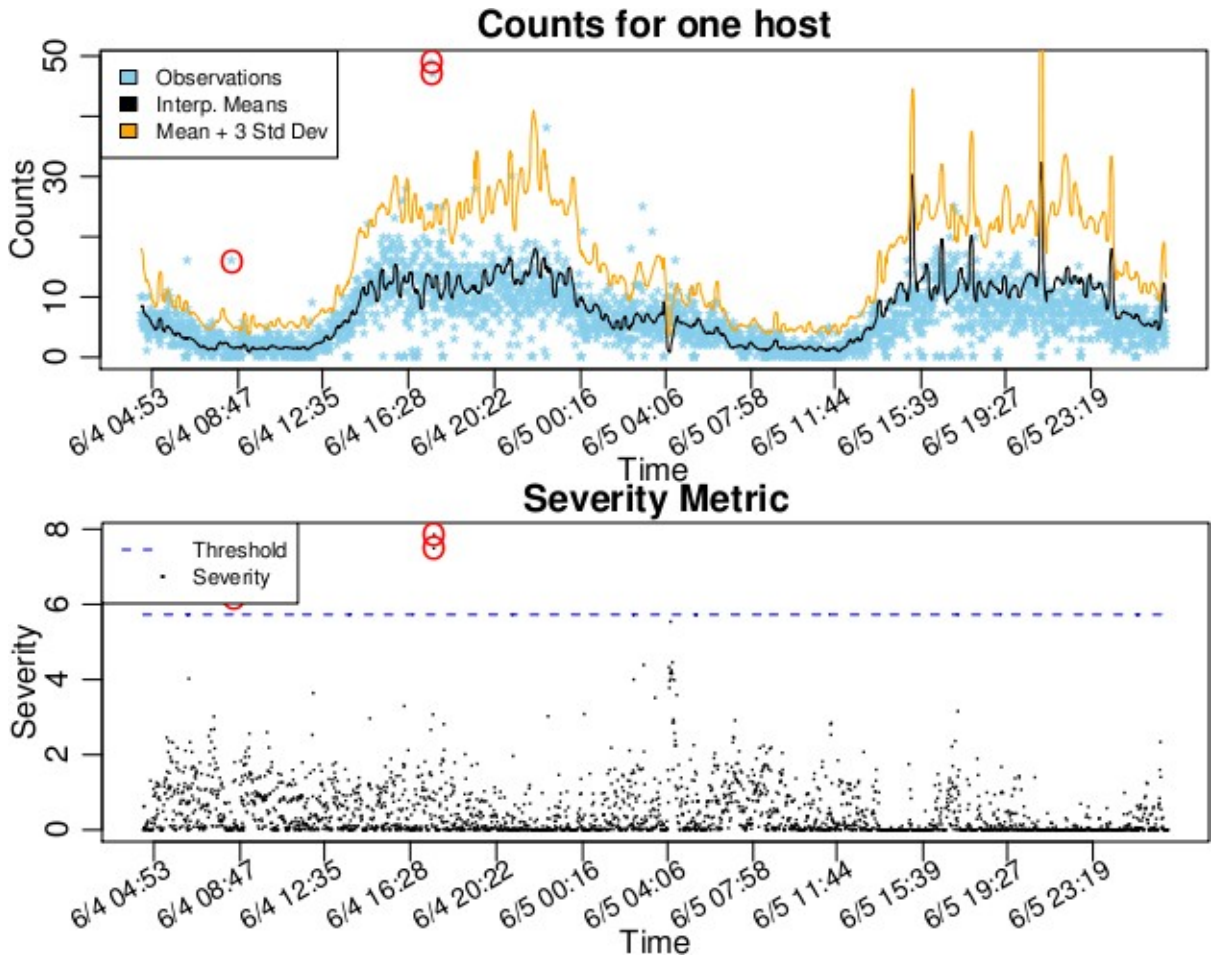
EMAAD Mark II

- We applied research by Lambert and Liu of Bell Labs [1]
  - Focuses on network degradation
  - Assumes a negative binomial model for changes
  - Also an EWMA
  - Quadratic mean interpolation
  - Has a normalized “Severity” metric
  - Outlier handling
    - Replace with random 'abnormal' draw
  - Able to detect gradual changes

EMAAD Mark II

- Modified to detect increases in activity
- Models 10 minute intervals independently
- Different cycles exist throughout the week
  - Monday-Thursday != Friday != Saturday,Sunday
  - More state than EMAAD Mk I
- Needs to train on a several cycles of data
  - Can't detect an abnormal Monday without seeing several normal ones first
  - Can use historical data, of which we have plenty

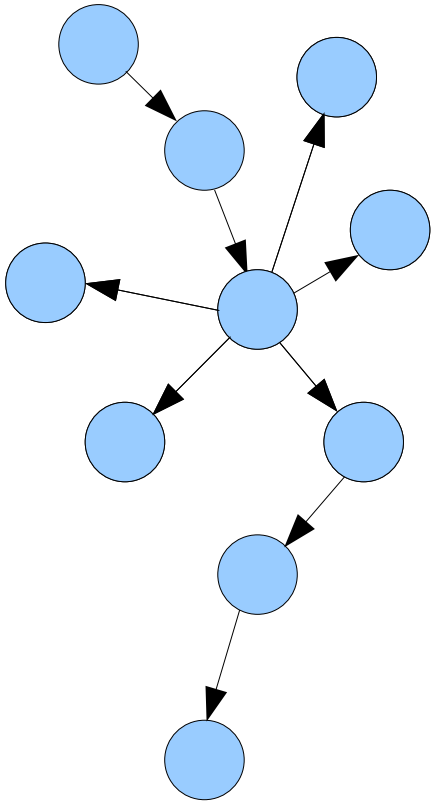
EMAAD Mark II



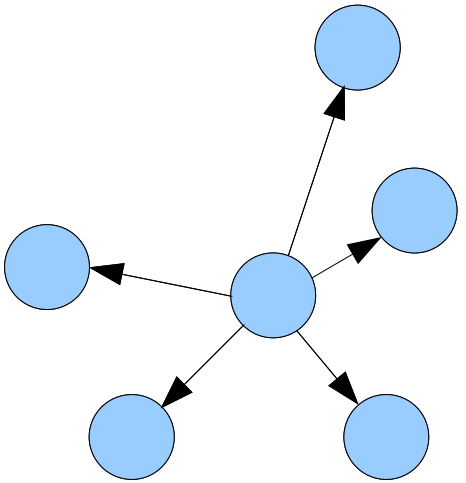
Next Generation Change Detection

- IP graphs based on flow records
  - Allows examination of subgraphs in a network
  - Looks for anomalies in a “neighborhood”
  - Assumes hackers behave locally
- Handles daily periods in the data
- Hidden Markov model used on edges of the graph
  - Distinct states of activity in flows
  - Probabilities in changing between states varies across time

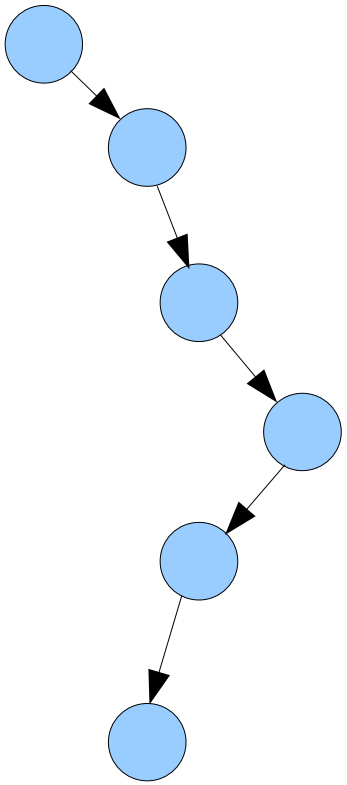
Next Generation Change Detection



Sample graph



Scan



Directed Attack

Response System

- Change detected, now what
  - Send e-mail to an analyst
  - Log this event somewhere
  - Quarantine a host
  - A combination of actions
- Not all events are equal
  - The responses shouldn't be either
  - A dynamic system needed

FRNSE

- Framework for Responding to Network Security Events
- “Replacing lowest level analyst work with a script”
- Takes alerts from Agents
  - Formated in XML
  - Transmitted securely
  - Alerts are queued and retried if unsuccessfully sent
- Uses rules to implement policy
- Can respond in a configurable way
- Exemptions can be specified
  - We still need Name Servers, pwn3d or not

FRNSE

- Python API for easy rule creation and response interfacing
- It has web interface for configuration
  - Rules
  - Exceptions
  - Data types
  - Viewing alerts

Agents

- EMAAD
- TippingPoint
- Snort
- AirDefense
- MassAV

FRNSE Production Results

- In 2007:
  - 919,737 alerts handled by FRNSE
  - 283,192 automatic firewall blocks
  - 2,293 analyst tickets generated
  - 179 automatic internal host quarantine events

Of 919,737 alerts generated in 2007, 99.75% were directly addressed by technical implementation of policy and required no analyst intervention.

Response Arsenal

- Internal Host Quarantine
- Perimeter block on external host
- Blackhole DNS name
- Open a Ticket for analysts
- Send email or page

Host-Based Quarantine

- Get a host off the network quickly
- Impact as few collateral systems as possible
- Handle the dynamic nature of a large network
  - Laptops move
  - Switches, ugh
- Deal with resourceful users
  - Jump ports
  - Change IP

Host-Based Quarantine

- Three lists need to be maintained by hand
  - The MAC vendor codes of switches
  - The IP addresses of routers
  - Every SNMP community string ever
- Harvest ARP tables from routers
- Switches found in ARP data
- Harvest forwarding tables from switches
- Crunch data
- Store in a database

Host-Based Quarantine

- Host location mapped and stored by MAC
- MAC/IP mappings stored in a separate table
  - MACs might map to multiple IPs, also stored
  - IPs could have multiple MACs, ugh
- Complete history* of every host's location
  - Useful in forensics/investigations
- Web interface
  - Can view a switch
  - Search by MAC or IP

Host-Based Quarantine

- Our current environment
  - Aprox. 1600 Switches, multiple vendors, vintages, settings
  - 20 Routers
- Runs every half hour
  - Parallelism in device harvesting and crunching needed to achieve this
  - Port jumpers handled after every run
  - Aprox. 40,000 database records generated per run
    - Aprox 2 million a day.
- Time from request to quarantine between 10 and 30 seconds depending on switch

Conclusion

- Flows used in change detection
- Detected changes feed into larger response system
- Of our responses we're pretty proud of Host-Based Quarantine

Questions

Alex Brugh

abrugh@lanl.gov

[1]D. Lambert and C. Liu. Adaptive Thresholds: Monitoring Streams of Network Counts Online. Journal of the American Statistical Association, 101(473):78–88, 2006.