Adaptive, Model-Based Monitoring And Threat Detection

Alfonso Valdes
Keith Skinner
SRI International
http://www.sdl.sri.com/emerald/adaptbn-paper/adaptbn.html

Outline

- **Objectives**
- **Approach**
  - Bayes net models
  - Key components: Session and availability monitors
  - TCP data characterization
  - What we detect
- **Results**
  - Llabs 99 data
  - EMERALD Live Demo Environment
  - Real World
- **Summary**

Objectives

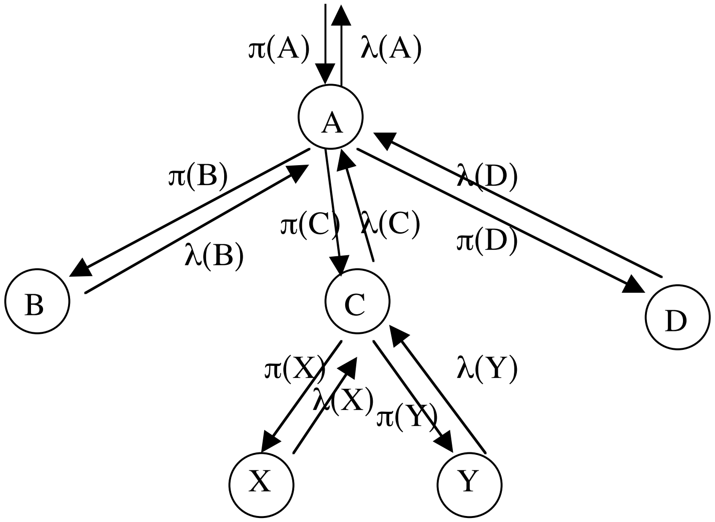
- **Explore middle ground between signature systems and anomaly detection**
- **Evaluate approach with data sets of interest**
  - Lincoln Labs data
  - Real-time demonstration environment
  - Real-world deployment
- **Establish:**
  - Generalization potential of important attack models
  - Ability to detect novel attacks

Approach

- **There is room for a detection paradigm that**
  - Comprehends attack models but
  - Reasons probabilistically
- **Bayes models seem like a good candidate**
  - We can describe or learn the statistical behavior of several observable variables under various modes of normal or attack behavior
- **Probabilistic aspect allows for generalization**
- **Our approach models normal and attack behaviors according to conditional probability tables**
- **Model-based aspect has multiple benefits:**
  - Superior to pure anomaly detection as far as threat classification
  - Models can be specified, learned, or hybrid
  - Capabilities beyond intrusion detection to resource availability monitoring

BN Algorithms

- **Describe the world in terms of conditional probabilities**
- **Model observables as nodes in a directed graph**
- **Children get π (prior) messages from parents**
- **Parents get λ (likelihood) messages from children**
  - At leaf nodes, λ messages correspond to observations
- **Belief state is updated as new evidence is observed**



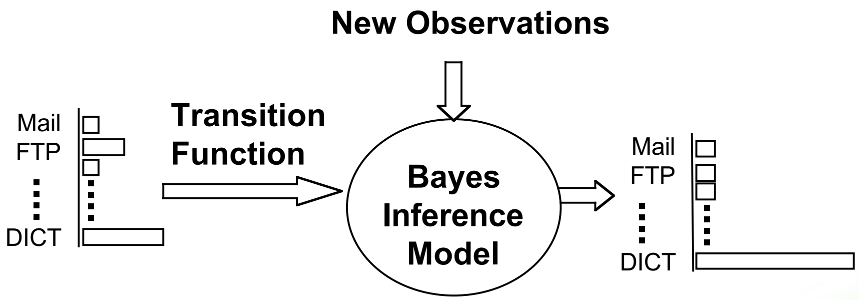
This diagram illustrates message propagation in a tree fragment

Learning, adaptation

- **Bayes models have a network structure and node parameters**
  - Conditional probability tables, or CPT
  - CPT(i,j)=P(child state = j | parent state = i)
- **We did not try to learn structure**
- **CPT's can be learned off-line or adaptively**
  - For real world data, no ground truth.
  - We observed "hypothesis capture" on very long runs
  - eBayes has optional capability to generate new hypotheses if no existing ones fit (resulted in discovery of unanticipated attacks in Lincoln data)
  - Stability of learning and hypothesis generation are still research issues for us
- **We have used offline learning to generate CPT's that perform well for the Lincoln data, the demo data, and real world data**

Transition and Update

- **New sessions start with a default prior over normal and attack hypotheses**
- **Inference results in new belief**
  - "In progress" alerts may be generated
- **This passes through a temporal transition model**
  - Tends to decay back to normal
  - But once a session is sufficiently suspicious, it will be reported
- **New inference results in updated belief**
- **Developing smarter transition model**



New Observations

Transition Function

Bayes Inference Model

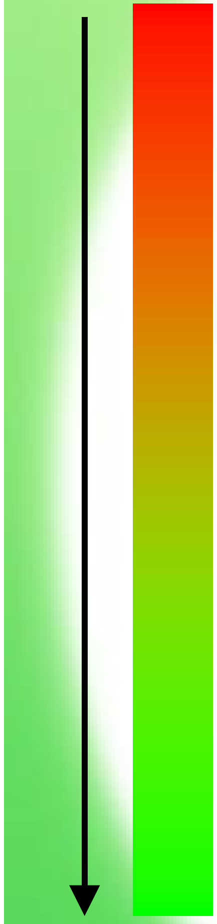
EMERALD Inference Techniques

- **Probabilistic systems can trigger on previously unseen patterns indicative of**
  - Suspicious activity
  - When things are heading south

	Anomaly Detection	Signature Engine	Bayes
Technique	Deviations from Learned Norms	Detect patterns of Interest	Probabilistic models of misuse
Generalization	Yes	May need new rules	Yes
Specificity	No	Yes	Yes
Sensitivity	Moderate	High	High
False alarms	Moderate	Low	Low
Adaptation	Yes	No	Yes

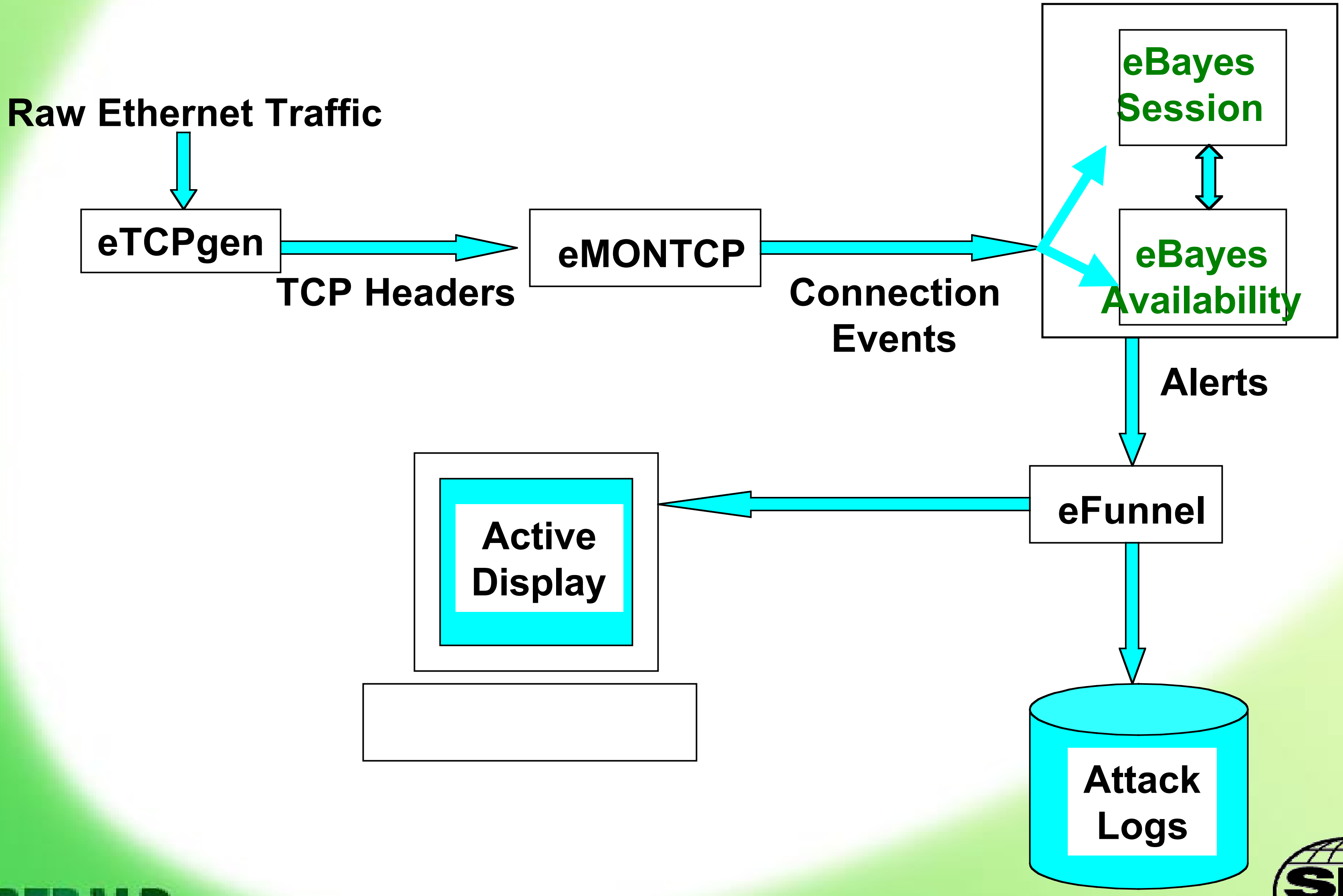
Key Components

Data Reduction



- **ETCPGEN processes live TCP traffic or TCPDUMP logs for batch experimentation and tuning**
  - Among other things, reassembles fragmented packets
  - Hardware pre-filtering (?)
- **EMONTCP processes ETCPGEN events**
  - Reconstructs TCP connections.
  - Adapts to traffic volume to estimate connection outcome
  - Also supplies source/destination address and port, connection setup time, data volume…
- **Session Monitor and Availability Monitor work concurrently from this point, using the same high-speed Bayes inference library**
- **Raw event rate reduced by a factor of 10^{-2}-10^{-3} at the output of EMONTCP**
- **Alerts are a small fraction of EMONTCP events**

eBayes Event Flow



TCP Data Characterization

- **At present, consider TCP headers, externally initiated connections to internal hosts only**
- **“Session” is a temporally contiguous burst from a source IP**
  - Session time out based on last event; whether there are any apparent open connections, etc.
  - Not too important to get exactly right (worst case: multiple alerts for the same attack session)
  - Considering random time out, longer for higher session “badness”
- **At the same time, valid hosts/ports are adaptively learned**
  - Accesses to invalid ports are considered more sensitive (detects stealth sweeps)
- **Component Correlation: the state of a service is communicated to the session monitor.**
  - If a service is down, prior expectation of certain error modes changes.
  - Alerts for “innocent victims” are largely suppressed
  - These are still part of the GUI report for the “service down” message (see below)

Detections: EBAYES Session Monitor

- **More of a conventional ID system, encodes important attack models in its conditional probabilitites**
- **Coupled to the availability monitor**
  - Prior expectation of anomalous session behavior conditioned on health of host/service requested

Attacks detected

- **Portsweeps (including stealthy sweeps of suspicious ports)**
- **IP Sweeps**
- **Floods: Syn floods, mail bombs, etc.**
- **Process table exhaustion**
- **Nonspecific high-error-rate traffic (often indicates password guessing)**
- **“Other BAD”**

Detections: Availability Monitor

- **EBAYES Availability Monitor (Blue Sensor)**
  - Dynamically learns valid traffic patterns via unsupervised discovery
  - Aging functions enable analysis of traffic bursts (response/recovery)
  - Bayes inference continuously gives a belief in service availability
  - Resolver alerts maintain threads of events. Outage resulting in millions of failure events are deinterleaved as to host, port, and clients.
  - Administrator sees a single report.

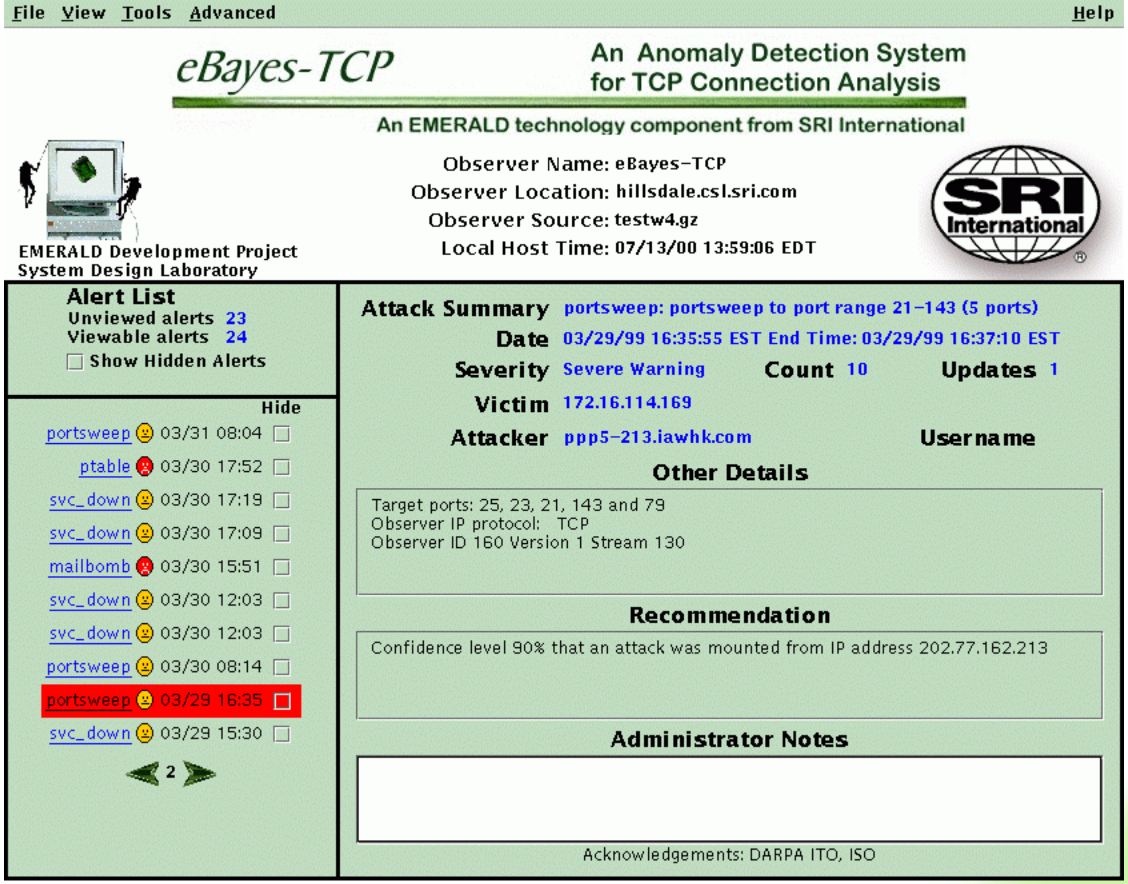
Capable of Adaptively Detecting

- **Excess failed connection rate**
- **Time to complete connection**
- **Variance from daily traffic norms**
- **Degraded state may or may not be due to an attack**

Lincoln Labs 99 Data

- **Detected 100% of visible Neptune (as syn flood)**
- **Detected all but 1 visible portsweep**
  - Naïve portsweeps trivial
  - Stealthy portsweeps detected based on accesses to invalid ports (“invalid” determined adaptively) - confidence usually lower
  - Missed portsweep was 4 ports from 3 different IP’s
- **Detected mailbombs**
- **Satans look like port or IP sweeps or syn floods**
- **Mscan looks like a portsweep and a syn flood**
- **Process table model covers process table and LL Apache attacks. Sucessful Apache also detected by availability monitor**
- **Detected several “dictionary”, “netcat”, and “selfping” attacks to various services WITH NO PREDEFINED MODEL**
- **Availability monitor detects, e.g., DOSNUKE**
- **No false alarms at 30% confidence threshold**

GUI Snapshot, LL week 4



eBayes-TCP

An Anomaly Detection System for TCP Connection Analysis

An EMERALD technology component from SRI International

EMERALD Development Project
System Design Laboratory

Observer Name: eBayes-TCP
Observer Location: hillsdale.csl.sri.com
Observer Source: testw4.gz
Local Host Time: 07/13/00 13:59:06 EDT

Alert List

Unviewed alerts 23
Viewable alerts 24
☐ Show Hidden Alerts

			Hide
portsweep	●	03/31 08:04	☐
ptable	●	03/30 17:52	☐
svc_down	●	03/30 17:19	☐
svc_down	●	03/30 17:09	☐
mailbomb	●	03/30 15:51	☐
svc_down	●	03/30 12:03	☐
svc_down	●	03/30 12:03	☐
portsweep	●	03/30 08:14	☐
portsweep	●	03/29 16:35	☐
svc_down	●	03/29 15:30	☐

◀ 2 ▶

Attack Summary portsweep: portsweep to port range 21-143 (5 ports)
Date 03/29/99 16:35:55 EST End Time: 03/29/99 16:37:10 EST
Severity Severe Warning **Count** 10 **Updates** 1
Victim 172.16.114.169
Attacker ppp5-213.iawhk.com **Username**

Other Details

Target ports: 25, 23, 21, 143 and 79
Observer IP protocol: TCP
Observer ID 160 Version 1 Stream 130

Recommendation

Confidence level 90% that an attack was mounted from IP address 202.77.162.213

Administrator Notes

Acknowledgements: DARPA ITO, ISO

EMERALD Live Demo Environment

- Live environment is a simulated e-commerce site behind a reasonably configured firewall
- Simulated normal traffic accesses allowed services
- A multi-stage attack is launched from a hacker console
- eBayes runs in integrated fashion with other EMERALD components
- Detects mscan (much stealthier than LL mscan: 28 connections, 6 ports, over in a flash)
- Detects syn flood
- Availability monitor detects success of syn flood
- Availability monitor detects physical disconnect
- Without modification, we detect nmap, strobe variants as portsweeps
- No false alarms at 30% confidence threshold

Real World

- We run this continuously monitoring our router to the outside
- Processes total about 15M, stable, and a few percent of CPU
  - 2M Packets/Day
  - 40K Connection events (synthetic)/Day
  - 4K Sessions/Day
  - ~20 Alerts/Day (Reduced by half via meta alert fusion - see my Thursday talk)
  - About 10 CPU minutes processing/Day, Pentium III/500, FreeBSD
- No ground truth
- Real traffic looks different:
  - New failure modes (added a “failed but innocuous http” model)
  - Traffic from robots and crawlers
- Nonetheless, we detect frequent IP sweeps. Details of some look like nasty known attacks
- Some apparent attempted syn_floods as well
- Detected down http (apparently non-malicious) before sysadmin

Real World Alerts

- **Observe about 20 alerts per day (1 per 200 sessions)**
- **Many are very likely good hits**
  - Sufficiently serious to get our sysadmin's attention
- **Many port 113 accesses**
  - Used by POP, IMAP, …
  - Filtered at the router, so appears invalid
  - Confidence usually around 35%
- **HTTP traffic with normal open/abnormal close connections**
  - New hypothesis generated, these largely go away
  - Looks like one of the LL 99 "Apache Back" attacks
- **Erroneous (but probably not malicious) DNS traffic**

Summary

- **Probabilistic model-based inference fills an important gap between anomaly detection and signature approaches**
- **We have a high-performance inference engine and two effective components**
  - TCP Session monitor
  - System availability monitor
- **Session monitor detects a variety of attacks in Lincoln data, demo data, and real data**
- **Key advantages of availability monitor:**
  - Dynamic discovery of resources or services (“did you know you had all those?”)
  - Real-time adaptation to traffic bursts
  - Rapid detection of degraded modes, due to attacks, coordinated attacks, or non-malicious faults