Two Complementary Views on Intrusion Detection

-- Macroscopic and Microscopic

Chin-Tser Huang

Department of Computer Science and Engineering
University of South Carolina

Network Activity

- Benign Traffic – Network traffic that should not result in a network compromise
  - – Web Browsing, E-mailing, etc.
- Malicious Traffic – Any activity intended to result in a compromise of a network entity
  - – Scanning, DoS, Session Hijacking, etc.

Computer Science and Engineering @ University of South Carolina

Network Intrusion Detection Systems

- Systems that look for malicious activities in a network environment
- Common classifications:
  - Signature/misuse-based
  - Anomaly-based
  - Hybrid

Signature/Misuse-Based Detection

- Attempts to fit malicious traffic characteristics to specific signatures
- Advantage
  - Very good at detecting known attacks
- Disadvantages
  - Can completely overlook novel attacks
  - Must constantly be updated

Denning's Assumption

- Malicious traffic is distinct from benign traffic
  - These differences are measurable
  - Example: Scanning has low probability of resulting in an established connection

Anomaly-Based Detection

- Treats benign traffic as norm
- Advantage
  - Can detect novel attacks
- Disadvantage
  - High false alarm rates
  - Costly computations

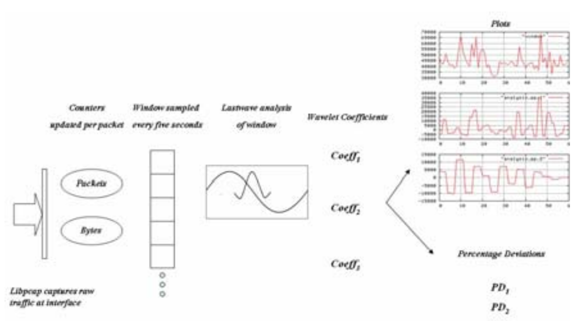
Computer Science and Engineering @ University of South Carolina

The Challenges

- How to keep the advantages of anomaly-based detection while reducing the false alarms?
- How to lower the overhead and detect anomalies in a timely fashion?
- How to automatically differentiate the detected anomalies?
- How to hold attacking hosts accountable?

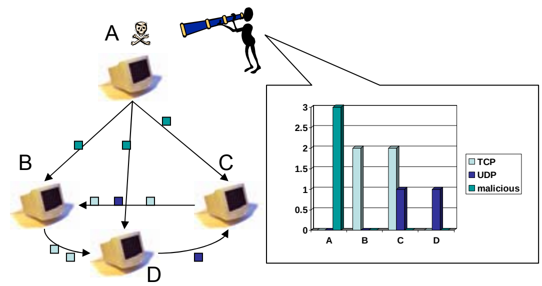
Two Complementary Views

- A ***macroscopic*** view
  - view network traffic as time-series signal
  - use wavelets to capture different types of anomalies



Two Complementary Views

- A ***microscopic*** view
  - view network as a collection of individual hosts
  - charge individual host for anomalous behavior

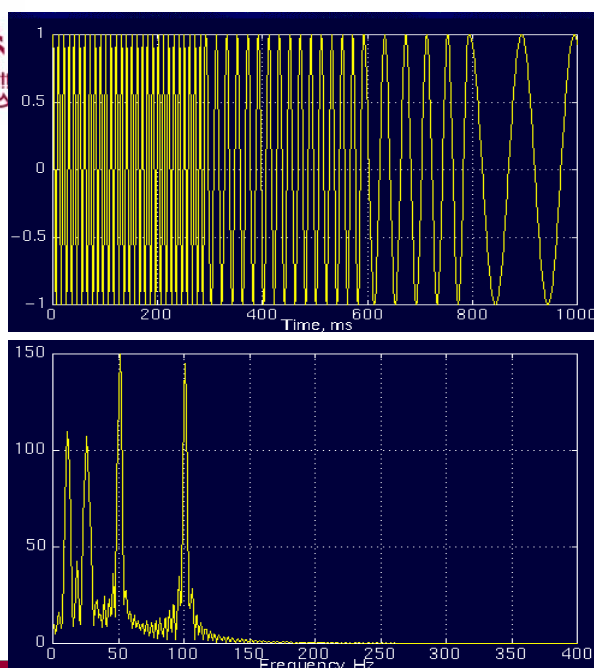


Macroscopic View

- Motivation
  - Perception at different detail levels, in close-to-real time
  - Applications include evaluation of security features, and for monitoring purposes
  - Build an Intrusion Detection System based on wavelet analysis

Macroscopic View

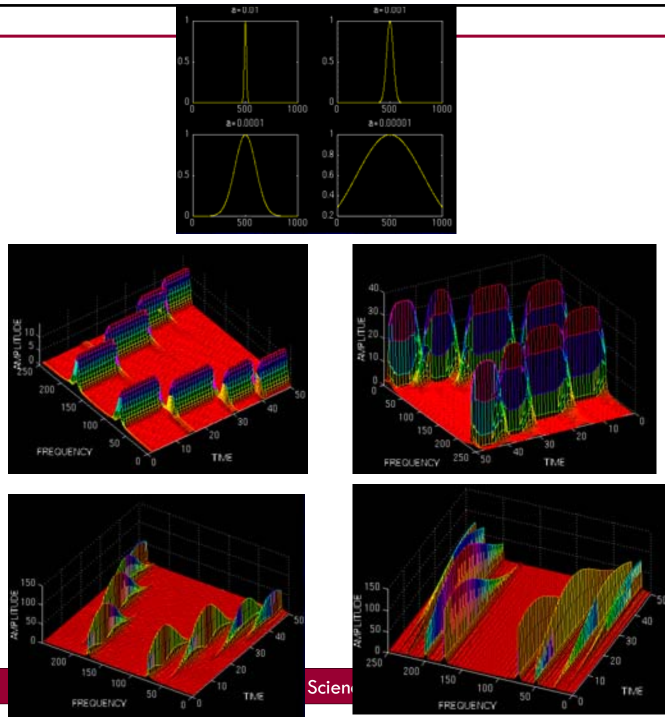
- Related works
  - “A Signal Analysis of Network Traffic Anomalies”, Paul Barford, Jeffery Kline, David Plonka and Amos Ron, ACM SIGCOMM Internet Measurement Workshop 2002
  - “A Wavelet-Based Approach to Detect Shared Congestion”, Min Sik Kim, Taekhyun Kim, Yong-June Shin, Simon S. Lam, and Edward J. Powers, ACM SIGCOMM 2004

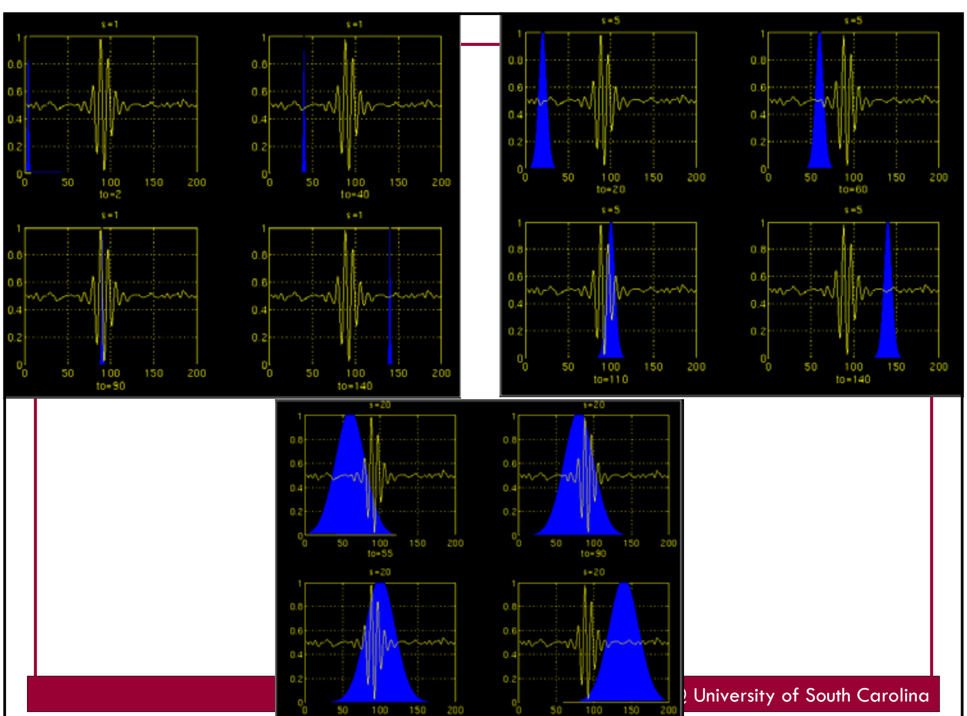


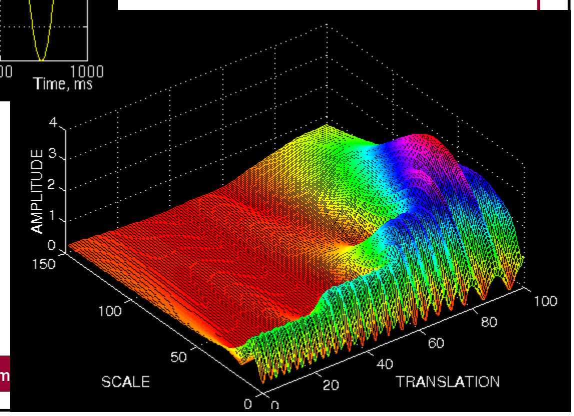
Frequencies?

Frequencies:

100, 50, 25, 10 Hz


a=0.01
a=0.001
a=0.0001
a=0.00001
AMPLITUDE
FREQUENCY
TIME

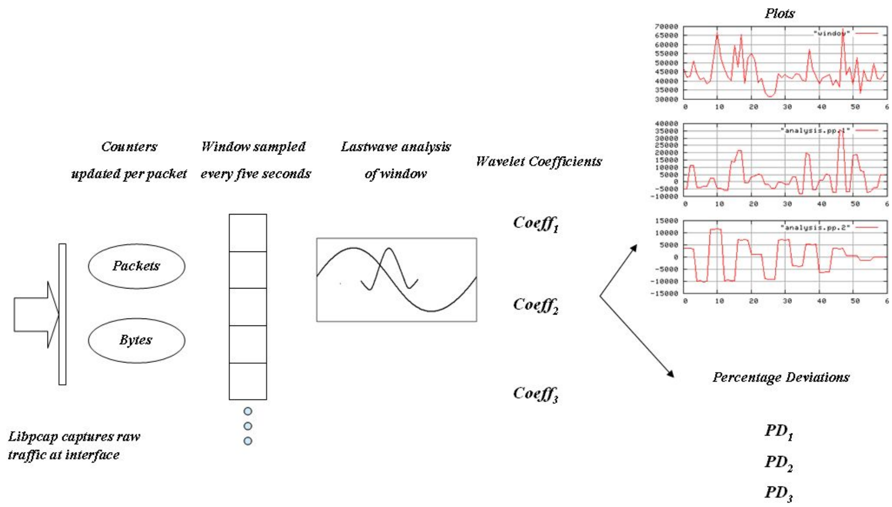

s=1
to=2
to=40
to=90
to=140
s=5
to=20
to=60
to=110
to=140
s=20
to=55
to=90
to=110
to=140



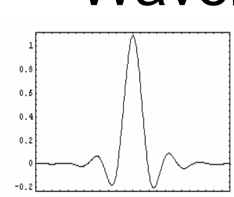
Analysis

- Iterative process **(Subband coding** or **Multi Resolution Analysis):**
  - Input for each iteration: a signal x of length N
  - Output: a collection of two, more derived signals, each of length $N/2$
  - Each output obtained by
    - convolving x with a specially designed filter F
    - decimating every other coefficient
    - $F(x)$ is the output signal
  - Special Filter L has a smoothing/averaging effect
    - corresponding output *low-frequency* output
  - Other filters, $H_1 \dots H_r$: discrete differentiation
    - output $H_i(x)$ should capture only the “fine-grained details”
  - Iterations proceed with the further decomposition of $L(x)$, creating the (shorter) signals $L^2(x);H_1L(x)\dots H_rL(x)$
- We obtain a family of output signals of the form $H_iL^{j-1}(x)$

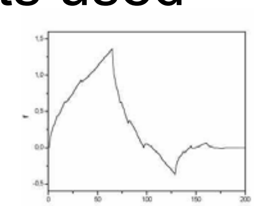
Framework



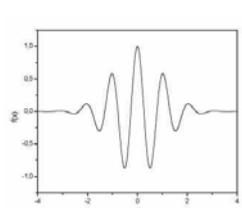
Wavelets used



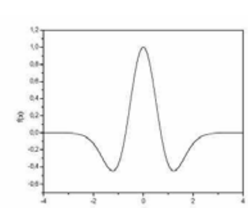
(a) Coiflet
Lengths: 11, 21, 41, 61



(b) Daubechies
Lengths: 6, 11, 21



(c) Morlet
Lengths: 15, 30, 40



(d) Mexican hat
Lengths: 15, 30, 40

Datasets

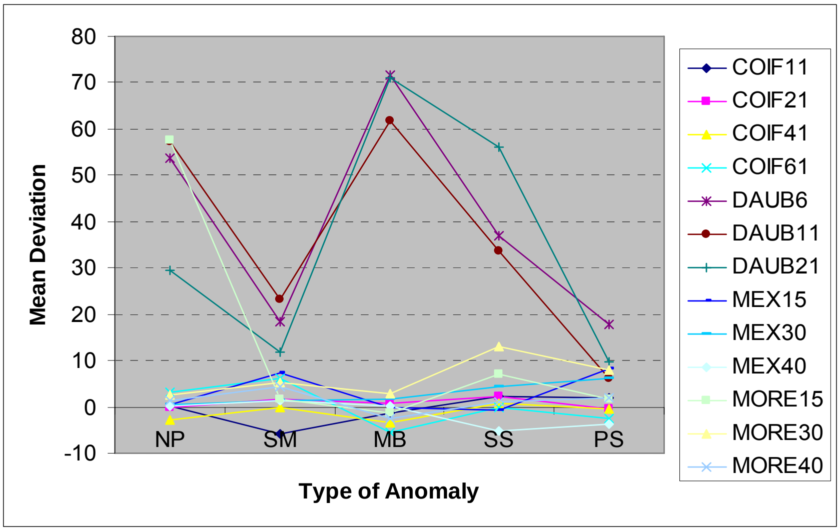
- MIT Lincoln Laboratory Intrusion Detection System Evaluation (1999)
  - Neptune
  - Smurf
  - Mailbomb
- EnetRegistry Inc. (2004-2005)
  - Portscan
  - Stealthscan

Evaluation

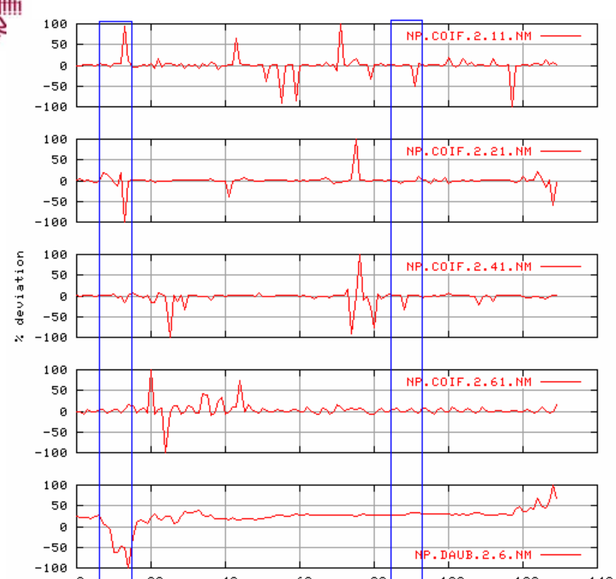
- Established anomalies
- Percentage Deviation: low value for the length of the anomaly is better
- Localization in time characteristics

Computer Science and Engineering @ University of South Carolina

Results: Deviation Characteristics

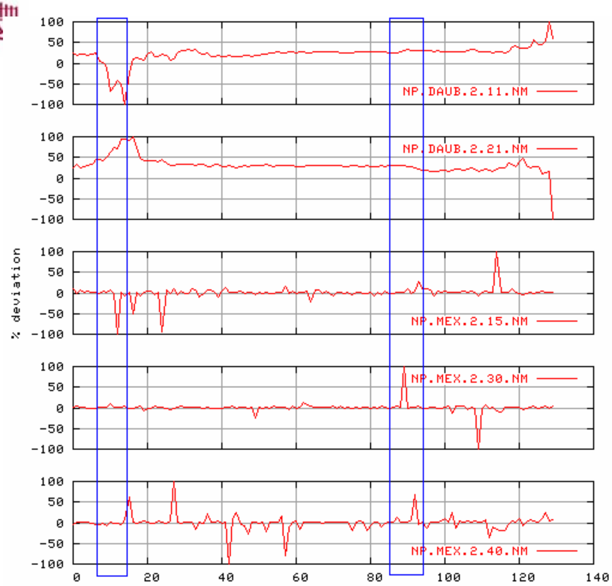


Results: Time Characteristics



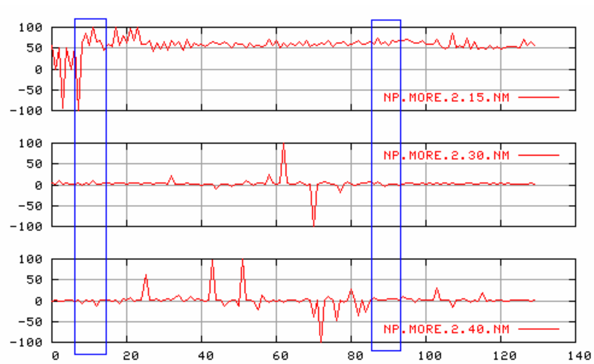
Localization in Time characteristics of Coiflet, Daubechies wavelets analyzed against Neptune attack

Results: Time Characteristics



Localization in Time characteristics of Daubechies, Mexican hat wavelets analyzed against Neptune attack

Results: Time Characteristics



Localization in Time characteristics of Morlet wavelets analyzed against Neptune attack

Results Summary

- Based on
  - Window length of five minutes
  - Lengths of filters,

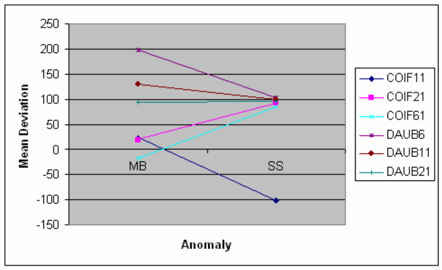
Coiflet wavelet and Mexican Hat wavelets show good characteristics for anomalies analyzed

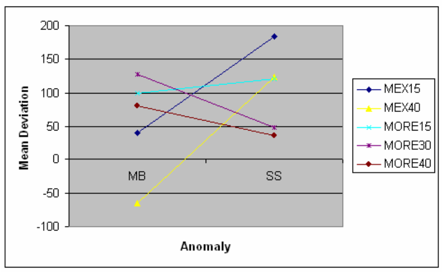
- Daubechies shows weakest characteristics for both localization in time and mean deviation

Next Step

- Varying window sizes
  - Anomalies are of varying sizes, need to be analyzed using different window sizes
- Other methods of evaluation
  - Entropy based
- Some preliminary results

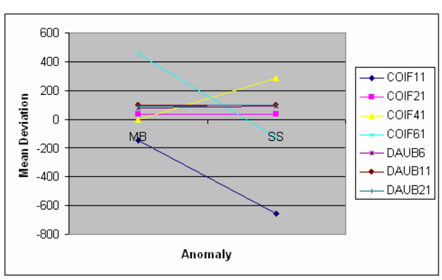
Varying Window Sizes

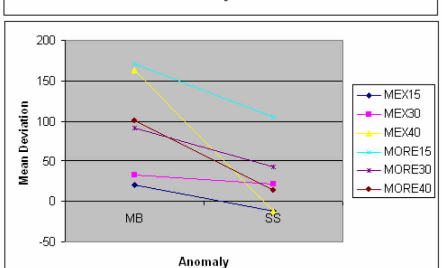




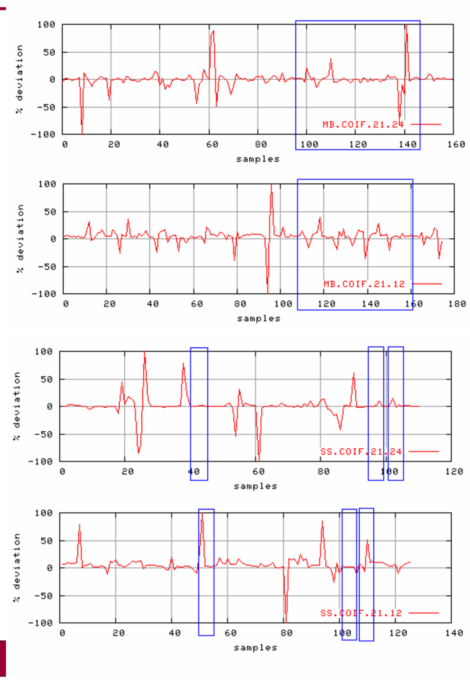
Mailbomb and Stealth scan anomalies analyzed using a window length of two minutes

Varying Window Sizes





Mailbomb and Stealth scan anomalies analyzed using a window length of one minute



a) Mailbomb, Coiflet, window lengths 24, 12

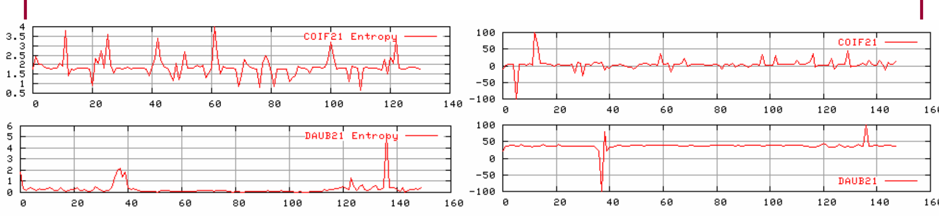
b) Stealth scan, Coiflet, window lengths 24, 12

Entropy Based Evaluation

Entropy: $H_r(x) = \frac{1}{1-r} log\left(\int f^r(x)dx\right),\qquad 0 < r < \infty, r \neq 1$

Rényi Entropy: $H_r(x) = -\frac{1}{2}\ \ln\left(\frac{1}{n}\ \sum_{i=1}^{n}\ f^r(n)\right)$

Neptune Attack, Coiflet and Daubechies Wavelets, window length one minute



Entropy Based

Percentage Deviation Based

Summary

- Real Time analysis
  - Generate signal from network traffic
  - Windowed analysis by subband coding/MRA
  - Evaluation of five anomalies from two datasets: low mean deviation, good localization in time
  - Coiflet and Mexican Hat wavelets show overall good characteristics, Daubechies shows poorest
- Implications:
  - Perception at different detail levels, in real time
  - Applications include evaluation of security features, and for monitoring purposes
  - Intrusion Detection System

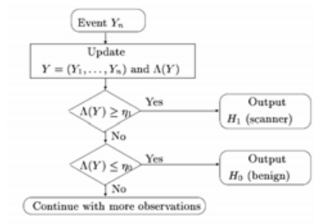
Microscopic View

- Motivation
  - Provide pinpointed analysis of anomalous activity at individual host
  - Keep computation overhead and memory consumption low
- Related works
  - Threshold Random Walk
  - Very Fast Containment of Scanning Worms

Threshold Random Walk

- Sequential hypothesis testing
  - Y=0 → success
  - Y=1 → failure
  - H_0= benign
  - H_1= malicious

$$\Lambda(Y) \equiv \frac{\Pr[Y|H_1]}{\Pr[Y|H_0]} = \Pi_{i=1}^{n} \frac{\Pr[Y_i|H_1]}{\Pr[Y_i|H_0]} \qquad (3)$$



Very Fast Containment of Scanning Worms

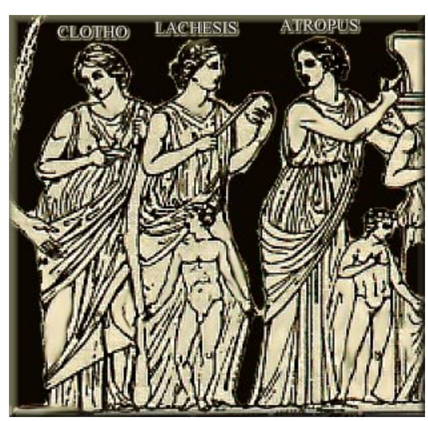
- A simplified version of TRW
- View the network as a collection of autonomous regions
- Uses approximation caches to limit memory consumption
- Counts the number of un-established connections

Fates

- Common features between Fates and both of these approaches
  - Granular view of the network
  - Examines state of connections
- Differences
  - Thresholds are dynamic
  - Charges are additive
  - Monitored hosts are always suspect

Fates Overview

- Three components
  - Clotho the Weaver
  - Lachesis the Apportioner
  - Atropos the Cutter of Threads



Fates Overview

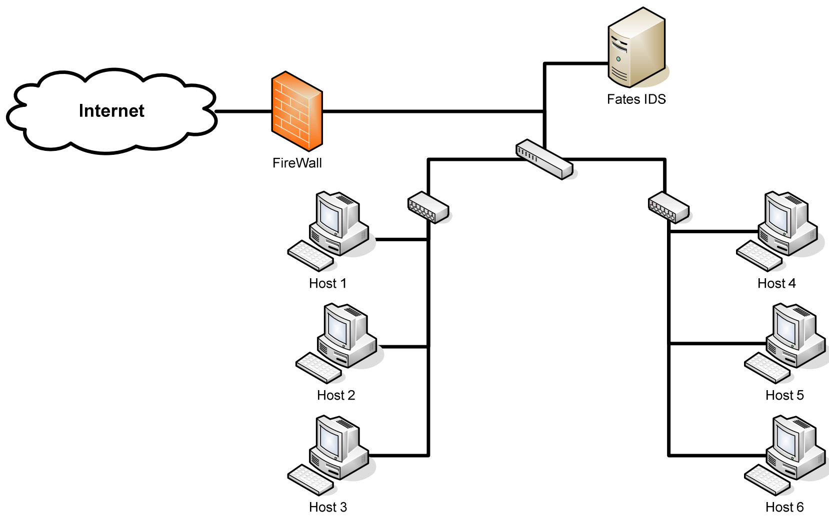
- Three components
  - Clotho the Weaver – Packet sniffer
    - Captures packets
  - Lachesis the Apportioner – Packet analyzer
    - Assesses charges to each host
  - Atropos the Cutter – Alarming mechanism
    - Produces human readable analysis

Sniffer

- Offline detection
  - Parsing TCPdump files of previously recorded traffic
- Real-time detection
  - Promiscuous capturing of packets as they come into/out of the network

Computer Science and Engineering @ University of South Carolina

Sniffer



Packet Processing

- The time of operation is divided into time steps (predefined by the user)
- Static windows are used to cut down on processing time
- All data used in analysis has a time-to-live measured in windows
  - Alleviates skewing of results

Packet Processing

- Maintains a list of internal IP addresses
- Two processing components
  - External Scan Detection Component
    - Detects scans from the outside world
  - Internal Host Monitor Component
    - Examines the state of monitored hosts' activities

Packet Processing

- External Scan Detection Component
  - Approximation cache of miss behavior
  - Provides a best approximation of potential scans with finite space requirements
  - If neither the source or destination is a monitored host, the packet could be part of a scan

Packet Processing

- Hash of the Source address is the index into an approximation cache
- TTL is set at each time step and whenever entry is accessed
- If count exceeds a threshold, the source is listed as a potential scanner

MAX_COUNT_TTL: 255
MAX_MISS_COUNT: 10
MAX_BLACKLIST_TTL: 255

Approximation Cache

Hash	Count	TTL
0	0	0
...	...	...
25	10	255
...	...	...
255	0	0

Black List

IP	TTL
192.168.1.100	255
...	...

Packet Processing

- Internal Host Monitor Component
  - Monitors subnet by IP or range of IP (stored in binary search tree)
    - A hash table of hosts
    - Current threshold
    - Current charge
  - Produces cumulative charges to be compared to individual thresholds

Packet Processing

- Each host is charged for each packet it sends
- Charge is a result of packet type
  - Connectionless
  - Connection-oriented

Packet Type	**Formula**
TCP	$\text{Charge} = 2 * (state - 1)$
UDP	$\text{Charge} = 2 * (count - 1)$

Packet Processing

- TCP state
  - Incoming packets decrease state by one
  - Outgoing packets increase state by one

	Type	Modifier
Incoming	SYN ACK FIN SYNACK FINACK	+1
Outgoing	SYN ACK FIN SYNACK FINACK RST	-1

Computer Science and Engineering @ University of South Carolina

Packet Processing

- UDP count
  - Number of packets with duplicate payload
  - Count of packet is stored in an approximation cache
    - Payload is hashed to index
    - Entries associated with a TTL

Packet Processing

- At end of time step
  - States used in TCP/IP are adjusted
    - If greater than zero, decremented by one
    - If less than zero, increased by one
  - TTL of elements in UDP's approximation cache is decremented by one
    - If TTL is zero, count is set to zero

Packet Processing

- At end of time step (continued)
  - All charges to hosts are added up
  - The total is compared to the host's initial threshold
    - Initial threshold is user defined for each host
    - If threshold is exceeded, threshold is set equal to the total

Packet Processing

- Threshold decay
  - If in subsequent time steps the average is less than the initial threshold, it is decayed
  - Average of time step charges
    - $avg = avg_{prev} * (1-\alpha) + TotalCharge * (\alpha)$

Packet Processing

- Threshold decay rate
  - $T_{current}=T_{current}-1/2(T_{initial} - avg)$
  - Quality:
    - Slowly redemptive
    - Decay rate is directly correlated to the history of a monitored host

Alarming

- In a well-behaved network the thresholds reach equilibrium
- In presence of scanning the threshold continually grows (only plateaus at saturation)
- This behavior is obvious upon observation (dependent on <u>human</u> interpretation)

Testing

- Experimental Data
  - Slammer (simulation)
    - Very effective worm
    - Blatantly obvious scanning behavior
  - Nmap (observed network traffic)
    - Standard issue scanning tool
    - Used to test TCP/IP detection capabilities

Testing

- Experimental Data (continued)
  - World of Warcraft (observed network traffic)
    - Sporadic packet transmission
    - Taxed servers with need for retransmission
  - Peer-to-Peer (observed network traffic)
    - Uses scanning to establish overlay network
    - Allows for file transfer

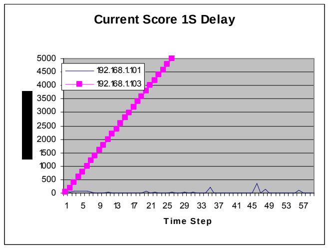
Slammer

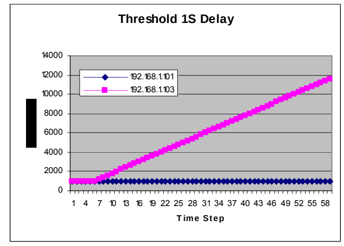
- High-speed worm
- Propagates through UDP packets
- Provides a good lower-bound

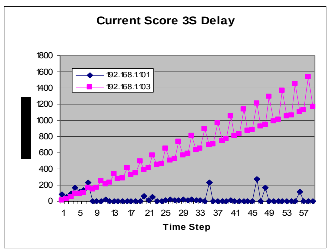
Slammer

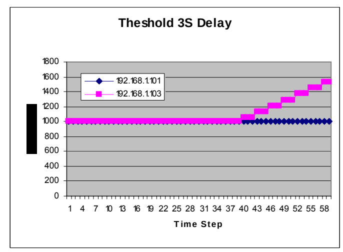
- Simulation
  - Advantages:
    - No legal issues
    - Specifics of the traffic are already known
    - Adjustable
  - Optional parameters:
    - Rate of Infection
    - Time of propagation
    - Size of network
    - Delay before inception of infection

Slammer









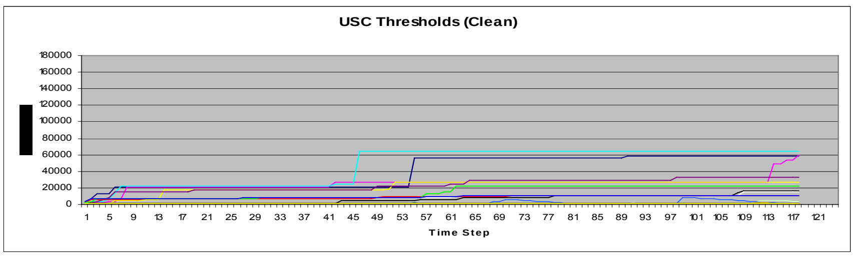
Nmap

- The network
  - Subset of the University of South Carolina's network
  - Monitoring 8 /24 subnets
  - Running Snort for comparison
- The scans
  - Half-Open scan
    - Also known as SYN scan
  - ACK scan
    - Distinct scan type
  - FIN scan
    - Stealth
  - RST scan
    - Stealth

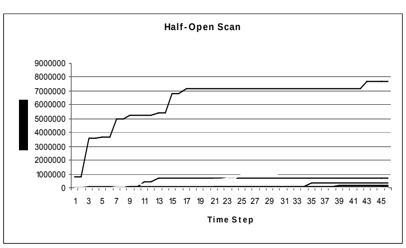
Nmap

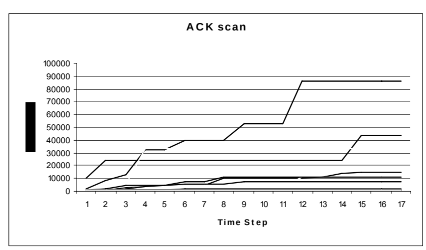
- Clean USC traffic
  - Thresholds tend to “jump” and “plateau”

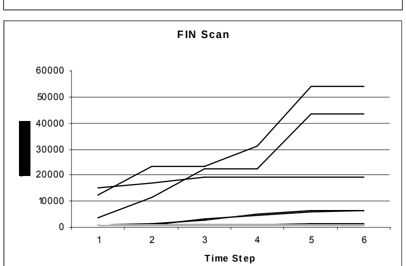
  - The network reaches equilibrium

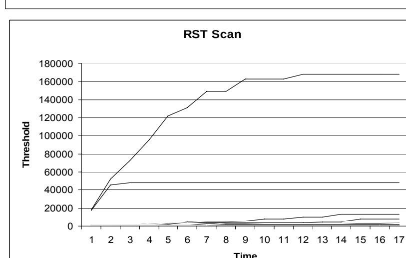


Nmap Thresholds









Computer Science and Engineering @ University of South Carolina

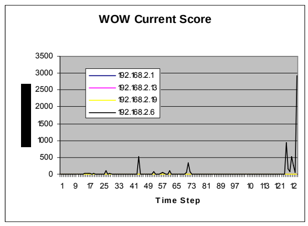
World of Warcraft

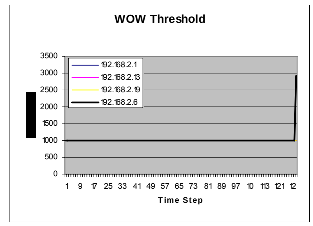
- Massively Multiplayer Online Role-Playing Game (MMORPG)
  - 1.5 million users
  - Several servers
    - Divided into regions
  - Possibility of lag due to congestion at servers

World of Warcraft

- TCPdump of 4 hosts on a home network
  - All ran HTTP traffic
  - One ran a WOW client
- Recorded 20 minutes of network traffic
  - Including: video streaming, HTTP, and WOW traffic

World of Warcraft





- The spikes are from transfer between servers
- Even in the presence of large lag, no extreme jumps in charges

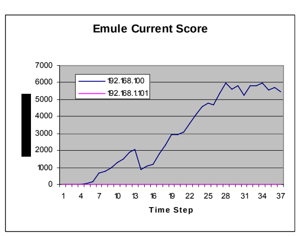
Peer-to-Peer Networks

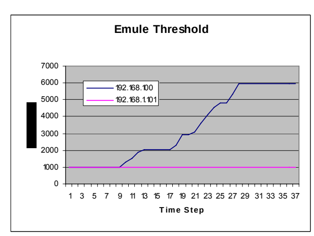
- Clients use scanning to find other peers, or contact a central servers
- Clients maintain a list of servers, but the server list changes
- Resembles scanning in finite space

Peer-to-Peer Networks

- Test data
  - TCPdump of Emule traffic from a home network
  - 1 host (no network activity)
  - 1 host running Emule client
    - Contacting servers
    - Transfer files

Peer-to-Peer Networks





- Though benign the attempt to connect to the servers resembles scanning
- As a result, the threshold looks similar to scans seen in the USC dataset

Evaluation

- Advantages
  - The simple calculations are still effective in detecting scans
  - Individual assessment of hosts aids in correcting the anomaly
  - Dynamic thresholds provide better understanding of diverse network entities

Evaluation

- Disadvantages
  - Does not distinguish between benign and malicious scanning
    - Intent is not our focus
  - Scalability
    - The less the granularity, the less the precision
  - Assumes source addresses are not spoofed
    - Many other such systems are also victim to this

Areas of Improvement

- Integrate a GUI interface
  - Alternately, integrate into other systems
- Integrate a rate of change analytical tool set
  - Providing automated alarming

Summary

- Fates provides a granular approach that allows for useful notification of anomalous activities
- Alarming is as specific as the user wishes
- Detection is feasible in a real-time network deployment without complex mathematical models

Conclusion

- Present two complementary views on intrusion detection
- Develop and implement two intrusion detection approaches based on the two views
- Experimental results show the effectiveness of the two approaches
- Investigate the feasibility of integration