Intrusion Detection Alert Flow Processing Using Time Series Analysis Methods

Jouni Viinikka
jouni.viinikka@orange-ftgroup.com
France Telecom / Supélec SSIR

2006-12-24
Sém. Diwall / ENST-Bretagne
• Processing a subset of alerts generated by Snort sensor
  - network-based
  - knowledge-based, signatures

• Operational environment, little choice over anything, notably
  - sensor type
  - sensor placement
  - sensor configuration and signatures

• Alert flood a known problem
  - In 1999 Manganaris et al. reported thousands of alerts per day per sensor, mostly false positives [MCZH99]
  - Julisch reported similar observations in 2001, 2002 and 2003 [Jul01, JD02, Jul03b, Jul03a]
Context(2): some reasons to the alert flood

- Sensor limitations
- Insufficient sensor configuration
- Increase in complementary use of sensor data
Flood causes(1): sensor limitations

- Unawareness of the operating environment
  - irrelevant positives
- Detection method limitations
  - false positives
- Mono-event analysis
  - the definition depends of the level of analysis
    - e.g. network packets vs. web server logs
  - limitations can be stretched
    - e.g. Snort tags and flowbits allow stateful detection
    - this remains modest
- Performance constraints limit algorithm complexity
Flood causes(2): insufficient sensor configuration

- Configuration increases awareness of the operating environment
  - can be difficult for technical and organizational reasons
    - dynamic environments
    - organizational and technical boundaries may not correspond
  - restricting analysis to packets from exterior to interior makes the sensor blind to internal activity, and attacks originating from internal networks
Flood causes(3): complementary uses

- Increase in complementary use of sensor data
  - monitoring normal system use and its functioning
    - SNMP, ICMP
    - performance measurement tools
  - less critical aspects of system policies
    - instant messaging
No good, operational solutions to alert flood

- Existing operational solutions are too simple
  - deactivate signatures
  - suppression of alerts related to certain hosts/networks
  - fixed thresholds

- Very limited
  - all or nothing at signature, network or host level, or
  - linear attenuation of volume
Proposed solutions for high priority and false alerts

- More sophisticated solutions proposed by researchers
  - sensor improvements
  - alert correlation (scenario recognition, verification, root cause analysis, ...)

- Eliminate/group/change alert priority
  - single alert's attributes
  - single alert's attributes + system configuration information
  - identified attack / false positive scenario

- No approaches for low severity true positives
Terminology(1) - alert flow analysis

- An **alert flow** is the sequence of alerts meeting the aggregation criteria.

- The **aggregation criteria** are the alert attribute(s) according to whose values a group of alerts can be formed.

- The **alert intensity** is the number of alerts in the alert flow during a **sampling interval**.

- The term **sampling** is used
  - in the signal processing sense: measurements taken at discrete intervals
  - not in the statistical sense: e.g. using 1 random packet out of sequence of $n$ packets (netflow)
Terminology(2) - alert flow analysis

- Let $y_t$ be the observed alert intensity at discrete time instant $t$. The **flow intensity** is the series of alert intensity observations $\{y_t\}$ for $t = 0, 1, 2, \ldots$.

- The **flow behavior** is the evolution of flow intensity as a function of time.
Terminology(3) - alert types

- True positive: correctly issued alert
  - with respect to alert description

- False positive: incorrectly issued alert
  - with respect to alert description

- Low impact positive: true positive
  - generally low severity
  - individually irrelevant
  - does not need immediate action from the operator
The four arguments of our thesis statement

1. Analysis of voluminous alert flows allows us to **extract useful information** when the significance of individual alerts cannot be determined.

2. It is possible that voluminous alert flows contain **significant regularities** related to the normal use of the system and its functioning.

3. It is **desirable and possible** to model these regularities.

4. These modeling techniques can be used to automatically
   
   (a) **filter out** the alerts related to normal flow behavior, and
   
   (b) **highlight** potentially interesting phenomena.
We address the first two arguments via real data

1. **Context:** intrusion detection and alert flood

2. **Thesis statement**

3. **Data:** collection, characteristics $\Rightarrow$ objective of modeling
   - collection: sites, sensors, signatures, storage
   - characteristics: low priority alerts, regular flows
   - two examples
   - objective of modeling and accompanying hypothesis

4. Three approaches to modeling normal behavior

5. Results and comparison of the three approaches

6. Conclusion and future work
Data collection

- Real environments
- Two production sites, A and B
  - Detailed in next two slides
- Three datasets
  - site A: set-1 and set-2
  - site B: set-3
- Snort sensors already in place
  - Default signatures
  - Few additional rules: hand-crafted, bleeding-snort
- Alerts logged into a database
- Analyzed post-mortem
Site A

- A production site without office workers
- Hosting services
Site B

- Office workers, some servers
- Services in computing center mostly used by site 2
1. Analysis of voluminous alert flows allows us to **extract useful information** when the significance of individual alerts cannot be determined.

2. It is possible that voluminous alert flows contain **significant regularities** related to the normal use of the system and its functioning.

3. It is **desirable and possible** to model these regularities.

4. These modeling techniques can be used to automatically
   (a) **filter** out the alerts related to normal flow behavior, and
   (b) **highlight** potentially interesting phenomena.
Flow characteristics after the analysis of the three sets

- **Low impact alerts**, triggering on
  - normal use of a system and its functioning
  - control and management traffic
  - applications whose use is tolerated/accepted
  - repetitive malicious phenomena
  - (possibly) back-scatter caused by (D)DoS attacks

- Often **impossible to determine significance of individual alert**
68% of the alerts in the set-1 are low impact

- Five most prolific signatures
- Total of 578 301 alerts over 43 days

<table>
<thead>
<tr>
<th>Signature</th>
<th># of alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP Request udp</td>
<td>176 009</td>
</tr>
<tr>
<td>ICMP PING WhatsupGold Windows</td>
<td>72 427</td>
</tr>
<tr>
<td>ICMP Destination Unreachable (Comm Adm Proh)</td>
<td>57 420</td>
</tr>
<tr>
<td>LOCAL-POLICY External connexion from HTTP server</td>
<td>51 674</td>
</tr>
<tr>
<td>ICMP PING Speedera</td>
<td>32 961</td>
</tr>
<tr>
<td></td>
<td>390 491</td>
</tr>
</tbody>
</table>
Monitoring a performance measurement tool

- Site A, ICMP PING speedera, alerts/hour, 32,961 in total
- Strong regularities
Monitoring a performance measurement tool

- Site A, ICMP PING speedera, alerts/hour, 32 961 in total
- Strong regularities
Monitoring a performance measurement tool

- Site A, ICMP PING speedera, alerts/hour, 32,961 in total
- Strong regularities
Monitoring a performance measurement tool

- Site A, ICMP PING speedera, alerts/hour, 32,961 in total
- Strong regularities, and potentially interesting anomalies
Monitoring a performance measurement tool

- Site A, ICMP PING speedera, alerts/hour, 32,961 in total
- Strong regularities, and potentially interesting anomalies
Monitoring a compromised host

- Hosted machine in the DMZ, no administrative access, must keep connected
- DDoS attack tool installed after a compromise
Monitoring a compromised host

- Hosted machine in the DMZ, no administrative access, must keep connected
- DDoS attack tool installed after a compromise
Monitoring a compromised host

- Site A, DDoS Stacheldraht agent -> handler (skillz)
- After initial identification, following alerts are low impact alerts
- Strong regularities
Monitoring a compromised host

- Site A, DDoS Stacheldraht agent $\rightarrow$ handler (skillz)
- After initial identification, following alerts are low impact alerts
- Strong regularities
Monitoring a compromised host

- Site A, DDoS Stacheldraht agent → handler (skillz)
- After initial identification, following alerts are low impact alerts
- Strong regularities and some potentially interesting anomalies
Monitoring a compromised host

- Site A, DDoS Stacheldraht agent -> handler (skillz)
- After initial identification, following alerts are low impact alerts
- Strong regularities and some potentially interesting anomalies
For various reasons, normal system use and behavior trigger low priority alerts:
- sensor limitations
- detection method limitations
- sensor usages

Because of the nature of the alerts:
- their volume can be large
- individual alerts are irrelevant
- flows can contain phenomena worth further investigation

Irrelevant alerts need to be filtered out:
- related work in alert correlation addresses either false alerts or high priority alerts
- need filtering methods which take into account the flow behavior
Filtering irrelevant alerts by using time series modeling

- **Assumption**: In low impact alert flows
  1. regular behavior is related to the normal use of the system, or
  2. only changes in any regular behavior are interesting

- **Model** such regularities, then use the model as a **filter**
  a) Free time for other tasks
  b) Highlight irregularities for the operator
Filtering irrelevant alerts by using time series modeling

- **Assumption**: In low impact alert flows
  1. regular behavior is related to the normal use of the system, or
  2. only changes in *any* regular behavior are interesting

- **Model** such regularities, then use the model as a filter
  a) Free time for other tasks
  b) Highlight irregularities for the operator
Modeling and filtering: how?

1. Context: intrusion detection and alert flood

2. Thesis statement

3. Data: collection, characteristics ⇒ objective of modeling

4. **Three approaches to modeling normal behavior**
   - exponentially weighted moving averages (EWMA)
   - stationary autoregressive (AR) models
   - non-stationary AR models

5. Results and comparison of the three approaches

6. Conclusion and future work
The modeling and experimentations to address 3 and 4

1. Analysis of voluminous alert flows allows us to extract useful information when the significance of individual alerts cannot be determined.

2. It is possible that voluminous alert flows contain significant regularities related to the normal use of the system and its functioning.

3. It is desirable and possible to model these regularities.

4. These modeling techniques can be used to automatically
   (a) filter out the alerts related to normal flow behavior, and
   (b) highlight potentially interesting phenomena.
Residual series contains anomalies

- Observed Intensity - Predicted Intensity = Model Residual

\[ y_t - \hat{y}_t = \tilde{y}_t \]

- Idea:
  - model predictions correspond to normal behavior
  - residual signal contains anomalous phenomena

- Idea and model not perfect
  - signal only most significant residual values to the operator
All models use previous observations to predict the current

\[
\hat{y}_t = \sum_{k=1}^{p} a^k_t y_{t-k} + e_t = \begin{bmatrix} a^1_t & a^2_t & \ldots & a^{p-1}_t & a^p_t \end{bmatrix} \begin{bmatrix} y_{t-1} \\ \vdots \\ y_{t-p} \end{bmatrix} + e_t,
\]

where \( \hat{y}_t \) is the predicted alert intensity at instant \( t \) and \( a^k_t \) is the \( k^{th} \) model coefficient at instant \( t \)

- **EWMA**
  - \( \hat{y}_t = \bar{y}_t = (1 - \lambda) \bar{y}_{t-1} + \lambda y_t, \ 0 < \lambda < 1 \)
  - \( p = \infty \)

- stationary AR: time-invariant coefficients \( a^k_t = a^k \)  

- Non-stationary autoregressive model (NAR) uses time-variant coefficients

\( \bar{y}_t = z_t \) of Chap. 4

simplifying approximation

Chap. 5

Chap. 6
Estimation of the model coefficients

- EWMA: no estimation
  - \( \{a^k\} \) decrease exponentially
  - \( a^k_t = a^k = \lambda (1 - \lambda)^{k-1} \)

- Stationary AR: least squares method
  - off-line process
  - training data

- NAR: Kalman filtering and fixed-lag smoothing algorithms
  - real-time data processing algorithms
  - model parameters are adapted dynamically
Non-stationary AR superior to two others

1. Context: intrusion detection and alert flood

2. Thesis statement

3. Data: collection, characteristics ⇒ objective of modeling

4. Three approaches to modeling normal behavior

5. Results and comparison of the three approaches

6. Conclusion and future work
The accuracy and consistency of models are different

- **Exponentially Weighed Moving Average (EWMA)**
  - the simplest and the lightest
  - very limited modeling capacity
  - works surprisingly well in practice for filtering
    - with our examples
    - performance can vary with the data set

- **Stationary autoregressive (AR) model**
  - slightly more complex model
  - difficult to interpret and inconsistent
  - limited applicability

- **Non-stationary AR model**
  - the same AR model
  - continuous estimation
  - accurate
  - consistent and easy to interpret
  - less dependent on data set than EWMA
EWMA works with stable flows
Stationary AR approach suffers from “flashbacks”
NAR approach very consistent compared to others.

Observations, 60 minute sampling interval, non-stationary AR(26) with kalman fixed lag smoother (1), 589233 alerts 11 anomalies.
NAR is clearly more accurate model than AR
The difference becomes from estimation algorithms

- AR model is estimated once
  - least squares algorithm
  - model coefficients minimize squared error in estimation data

- NAR model coefficients are estimated dynamically
  - Kalman filter / smoother algorithms
  - adaptive model
  - efficient computation
Classical least squares estimation for AR

- We minimize the sum of squared error $\|r^2\|$ over all estimation data of length $T$

- We note

$$Y_t = H\theta + r$$

$$\begin{bmatrix} y_{p+1} \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} y_p & \cdots & y_1 \\ \vdots & \ddots & \vdots \\ y_{T-1} & \cdots & y_{T-p} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix} + \begin{bmatrix} r_{p+1} \\ \vdots \\ r_T \end{bmatrix}$$

- $\|r^2\|$ minimized with

$$\hat{\theta}_{LS} = (H^TH)^{-1} H^T Y_t$$
Kalman filter: state space formalism and equations

- $C_\theta$ for the covariance matrix of $\theta$

- $\theta_t = (-a_1^t \ldots - a_p^t)^T$ and $H_t = (y_{t-1} \ldots y_{t-p})$

- Using random walk evolution, state and observation equations are

  $\theta_{t+1} = \theta_t + w_t$

  $y_t = H_t \theta_t + e_t$

- Kalman filter equations to update $\hat{\theta}_t$ with every new $y_t$

  $C_{\tilde{\theta}_t|t-1} = C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}}$

  $K_t = C_{\tilde{\theta}_t|t-1} H_t^T (H_t C_{\tilde{\theta}_t|t-1} H_t^T + C_{e_t})^{-1}$

  $\epsilon_t = y_t - H_t \hat{\theta}_{t-1}$

  $\hat{\theta}_t = \hat{\theta}_{t-1} + K_t \epsilon_t$

  $C_{\tilde{\theta}_t} = (I - K_t H_t) C_{\tilde{\theta}_t|t-1}$
Kalman filter: state space formalism and equations

- $C_{\theta}$ for the covariance matrix of $\theta$
- $\theta_t = (-a_1^t \ldots - a_p^t)^T$ and $H_t = (y_{t-1} \ldots y_{t-p})$
- Using random walk evolution, state and observation equations are
  \[
  \theta_{t+1} = \theta_t + w_t \\
  y_t = H_t \theta_t + e_t
  \]
- Kalman filter equations to update $\hat{\theta}_t$ with every new $y_t$
  \[
  C_{\tilde{\theta}_{t\mid t-1}} = C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}} , \\
  K_t = C_{\tilde{\theta}_{t\mid t-1}} H_t^T (H_t C_{\tilde{\theta}_{t\mid t-1}} H_t^T + C_{e_t})^{-1} , \\
  \epsilon_t = y_t - H_t \hat{\theta}_{t-1} , \\
  \hat{\theta}_t = \hat{\theta}_{t-1} + K_t \epsilon_t , \\
  C_{\tilde{\theta}_t} = (I - K_t H_t) C_{\tilde{\theta}_{t\mid t-1}} .
  \]
Kalman filter: state space formalism and equations

- $C_\theta$ for the covariance matrix of $\theta$
- $\theta_t = (-a_1^t \ldots - a_p^t)^T$ and $H_t = (y_{t-1} \ldots y_{t-p})$
- Using random walk evolution, state and observation equations are

$$\theta_{t+1} = \theta_t + w_t$$
$$y_t = H_t \theta_t + e_t$$

- Kalman filter equations to update $\hat{\theta}_t$ with every new $y_t$

$$C_{\tilde{\theta}_t|t-1} = C_{\tilde{\theta}_{t-1}} + C_{w_{t-1}}$$
$$K_t = C_{\tilde{\theta}_t|t-1} H_t^T \left( H_t C_{\tilde{\theta}_t|t-1} H_t^T + C_{e_t} \right)^{-1}$$
$$\epsilon_t = y_t - H_t \hat{\theta}_{t-1}$$
$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t \epsilon_t$$
$$C_{\tilde{\theta}_t} = (I - K_t H_t) C_{\tilde{\theta}_{t|t-1}}$$
Kalman filter: state space formalism and equations

- $C\theta$ for the covariance matrix of $\theta$

- $\theta_t = (-a_1^t \ldots - a_p^t)^T$ and $H_t = (y_{t-1} \ldots y_{t-p})$

- Using random walk evolution, state and observation equations are

$$\theta_{t+1} = \theta_t + w_t$$
$$y_t = H_t \theta_t + e_t$$

- Kalman filter equations to update $\hat{\theta}_t$ with every new $y_t$

$$C_{\hat{\theta}_t|t-1} = C_{\hat{\theta}_t|t-1} + C_{w_{t-1}} ,$$
$$K_t = C_{\hat{\theta}_t|t-1} H_t^T (H_t C_{\hat{\theta}_t|t-1} H_t^T + C_{e_t})^{-1} ,$$
$$\epsilon_t = y_t - H_t \hat{\theta}_{t-1} ,$$
$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t \epsilon_t ,$$
$$C_{\hat{\theta}_t} = (I - K_t H_t) C_{\hat{\theta}_t|t-1} .$$
## NAR applied for set-3

<table>
<thead>
<tr>
<th>Flow</th>
<th>Observ ($10^3$)</th>
<th>Alerts ($10^6$)</th>
<th>An</th>
<th>Prop ($10^{-3}$)</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP request</td>
<td>13</td>
<td>3.0</td>
<td>362</td>
<td>28</td>
<td>NOK</td>
</tr>
<tr>
<td>SNMP public</td>
<td>13</td>
<td>2.8</td>
<td>16</td>
<td>1.2</td>
<td>OK</td>
</tr>
<tr>
<td>L3retriever</td>
<td>13</td>
<td>1.3</td>
<td>39</td>
<td>3.0</td>
<td>OK</td>
</tr>
<tr>
<td>(http.inspect)</td>
<td>13</td>
<td>0.4</td>
<td>147</td>
<td>11</td>
<td>NOK</td>
</tr>
<tr>
<td>DCERPC 1</td>
<td>13</td>
<td>0.4</td>
<td>15</td>
<td>1.2</td>
<td>OK</td>
</tr>
<tr>
<td>DCERPC 2</td>
<td>5</td>
<td>0.1</td>
<td>14</td>
<td>2.8</td>
<td>OK</td>
</tr>
<tr>
<td>SMB-DS IPC$ 1</td>
<td>13</td>
<td>0.4</td>
<td>24</td>
<td>1.8</td>
<td>OK</td>
</tr>
<tr>
<td>SMB-DS IPC$ 2</td>
<td>4</td>
<td>0.2</td>
<td>9</td>
<td>2.3</td>
<td>OK</td>
</tr>
<tr>
<td>SMB IPC$</td>
<td>13</td>
<td>0.3</td>
<td>22</td>
<td>1.7</td>
<td>OK</td>
</tr>
</tbody>
</table>

- 13 000 observations with $t_s = 1$ min $\sim 9$ days
- Significant reduction in alerts and intervals reported to the operator
NAR applied for set-3

<table>
<thead>
<tr>
<th>Flow</th>
<th>Observ (10^3)</th>
<th>Alerts (10^6)</th>
<th>An</th>
<th>Prop (10^{-3})</th>
<th>OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNMP request</td>
<td>13</td>
<td>3.0</td>
<td>362</td>
<td>28</td>
<td>NOK</td>
</tr>
<tr>
<td>SNMP public</td>
<td>13</td>
<td>2.8</td>
<td>16</td>
<td>1.2</td>
<td>OK</td>
</tr>
<tr>
<td>L3retriever</td>
<td>13</td>
<td>1.3</td>
<td>39</td>
<td>3.0</td>
<td>OK</td>
</tr>
<tr>
<td>(http.inspect)</td>
<td>13</td>
<td>0.4</td>
<td>147</td>
<td>11</td>
<td>NOK</td>
</tr>
<tr>
<td>DCERPC 1</td>
<td>13</td>
<td>0.4</td>
<td>15</td>
<td>1.2</td>
<td>OK</td>
</tr>
<tr>
<td>DCERPC 2</td>
<td>5</td>
<td>0.1</td>
<td>14</td>
<td>2.8</td>
<td>OK</td>
</tr>
<tr>
<td>SMB-DS IPC$ 1</td>
<td>13</td>
<td>0.4</td>
<td>24</td>
<td>1.8</td>
<td>OK</td>
</tr>
<tr>
<td>SMB-DS IPC$ 2</td>
<td>4</td>
<td>0.2</td>
<td>9</td>
<td>2.3</td>
<td>OK</td>
</tr>
<tr>
<td>SMB IPC$</td>
<td>13</td>
<td>0.3</td>
<td>22</td>
<td>1.7</td>
<td>OK</td>
</tr>
</tbody>
</table>

- 13 000 observations with $t_s = 1$ min $\sim 9$ days
- Significant reduction in alerts and intervals reported to the operator
Use NAR if possible

- Which model to use?
  - NAR whenever computational resources allow
    - accurate
    - consistent in highlighting rapid changes
    - the limitation of AR model protect against risks of dynamic estimation
  - EWMA when very efficient computation is required, and flows rather stable
  - Both models can be used with one set of parameters
    - Tradeoff ease-of-deployment vs. model accuracy
Revisiting thesis statement

1. Context: intrusion detection and alert flood

2. Thesis statement

3. Data: collection, characteristics ⇒ objective of modeling

4. Three approaches to modeling normal behavior

5. Results and comparison of the three approaches

6. Conclusion and future work
Chronological summary(1)

- The problem arose from the real data: a need for flow based processing

- We started as simple as possible: trend modeling with EWMA [VD04]
  - despite the model’s limitations, it worked rather well rather often
  - implemented as a component of the SIM platform used at France Telecom [DV05, DV06]

- To address the limitations of EWMA model, stationary AR models were a natural next step towards more complex models [VDMS06]
  - desirable property: incapable to model abrupt changes
  - signal transforms: inconsistent results, difficult to interpret

- Non-stationary AR models allowed us to
  - keep the desired properties
  - leave the signal transforms behind
  - a publication being written
• The stationary AR and non-stationary AR approaches are implemented using Matlab
  - not at product stage
  - specifications could be written based on the results of the thesis

• The product aspect was not included in original plan, but is a positive result
The four arguments of our thesis statement

1. Analysis of voluminous alert flows allows us to extract useful information when the significance of individual alerts cannot be determined.  
   Data analysis

2. It is possible that voluminous alert flows contain significant regularities related to the normal use of the system and its functioning.  
   Data analysis

3. It is desirable and possible to model these regularities.  
   Data analysis, NAR

4. These modeling techniques can be used to automatically
   (a) filter out the alerts related to normal flow behavior, and  
       EWMA, NAR
   (b) highlight potentially interesting phenomena.  
       EWMA, NAR
Future work (1)

- **Spectral analysis**
  - The coefficients of NAR model can be used to estimate time-varying power spectral density
  - Regular components could be easier to recognize and isolate from time-frequency representation
  - Abrupt changes should be visible throughout the spectrum

- **3-scale vs. multi-scale analysis**
  - The sampling interval defines the time frequency resolution, three different sampling intervals allow to widen the resolution slightly
  - Multi-scale analysis such as wavelet analysis is more complete
  - What is the gain with respect to computational cost
Future work (2)

- Combining filtering with root cause analysis
  - Finer-grained aggregation criteria

- Invariant measures
  - Analysis of other metrics than alert intensity
  - If finding metrics that remain invariant during normal use, the anomaly detection could be simplified

- Pushing analysis techniques to sensors
  - Can we combine several flows, or do they need be analyzed separately?
  - Fast implementation in HW
Intrusion Detection Alert Flow Processing Using Time Series Analysis Methods

Jouni Viinikka

jouni.viinikka@orange-ftgroup.com

France Telecom / Supélec SSIR

2006-12-24

Sém. Diwall / ENST-Bretagne
References


