Analysis of FRAUD network ACTIONS; rules and models for detecting fraud activities
FRAUD ?

• HACKERS !!
  – *DoS*: Denial of service
  – *R2L*: Unauth. Access
  – *U2R*: Root access to Local Machine.
  – *Probing*: Survallience.
  – ....

  First PC Viruses: *Prophets of PC Viruses*

• Our Case: **Fraud = Anomally**
EXPECTATIONs?

- concise data?
  (Feature Weightining and Selection)

- model for classification?
  (Model Extration)

- what indicates fraud?
  (Rule extraction)
Constraints

DATA

- HUGE
- UNBALANCED
- HIGH DIMENSIONAL
- NOMINAL+NUMERICAL FEATs
- DIVERGENT CASES

MACHINE

- MEMORY
- TIME
DATA

- **KDD'99** Data
  - TCP dump on LAN -> Lincoln Labs-MIT **
  - Peppered with attacks.
  - 4 million records (reduction need)
  - 42 Features (selection need)
  - 16 Attack Type (merge as anomaly)

*Deficient for detecting content based attacks.*


** http://www.ll.mit.edu/
Instances

• **Instance** – 2 sec snapshots of **connection**.
  – **Connection** – TCP packages between same two IPs

• Each instance ~ 80 bytes
43 features

- **Categories**
  - Host
  - Service (Telnet, Voip)
  - Content
    - Exp. Root access, Shell activation
  - TCP stats
  - Destination host

- 4 nominal vs 39 numeric features
TCP stats

<table>
<thead>
<tr>
<th>feature name</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>duration</td>
<td>continuous</td>
</tr>
<tr>
<td>protocol_type</td>
<td>discrete</td>
</tr>
<tr>
<td>service</td>
<td>discrete</td>
</tr>
<tr>
<td>src_bytes</td>
<td>continuous</td>
</tr>
<tr>
<td>dst_bytes</td>
<td>continuous</td>
</tr>
<tr>
<td>flag</td>
<td>discrete</td>
</tr>
<tr>
<td>land</td>
<td>discrete</td>
</tr>
<tr>
<td>wrong_fragment</td>
<td>continuous</td>
</tr>
<tr>
<td>urgent</td>
<td>continuous</td>
</tr>
</tbody>
</table>
## Content Features

<table>
<thead>
<tr>
<th>feature name</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>continuous</td>
</tr>
<tr>
<td>num_failed_logins</td>
<td>continuous</td>
</tr>
<tr>
<td>logged_in</td>
<td>discrete</td>
</tr>
<tr>
<td>num_compromised</td>
<td>continuous</td>
</tr>
<tr>
<td>root_shell</td>
<td>discrete</td>
</tr>
<tr>
<td>su_attempted</td>
<td>discrete</td>
</tr>
<tr>
<td>num_root</td>
<td>continuous</td>
</tr>
<tr>
<td>num_file_creations</td>
<td>continuous</td>
</tr>
<tr>
<td>num_shells</td>
<td>continuous</td>
</tr>
<tr>
<td>num_access_files</td>
<td>continuous</td>
</tr>
<tr>
<td>num_outbound_cmds</td>
<td>continuous</td>
</tr>
<tr>
<td>is_hot_login</td>
<td>discrete</td>
</tr>
<tr>
<td>is_guest_login</td>
<td>discrete</td>
</tr>
</tbody>
</table>
Same Host & Same Service

<table>
<thead>
<tr>
<th>feature name</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>continuous</td>
</tr>
<tr>
<td>serror_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>rerror_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>same_srv_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>diff_srv_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>srv_count</td>
<td>continuous</td>
</tr>
<tr>
<td>srv_serror_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>srv_rerror_rate</td>
<td>continuous</td>
</tr>
<tr>
<td>srv_diff_host_rate</td>
<td>continuous</td>
</tr>
</tbody>
</table>
Destination Host

@attribute 'dst_host_count' real
@attribute 'dst_host_srv_count' real
@attribute 'dst_host_same_srv_rate' real
@attribute 'dst_host_diff_srv_rate' real
@attribute 'dst_host_same_src_port_rate' real
@attribute 'dst_host_srv_diff_host_rate' real
@attribute 'dst_host_serror_rate' real
@attribute 'dst_host_srv_serror_rate' real
@attribute 'dst_host_rerror_rate' real
@attribute 'dst_host_srv_rerror_rate' real
@attribute 'dst_host_srv_rerror_rate' real
All List

- duration: continuous.
- protocol_type: symbolic.
- service: symbolic.
- flag: symbolic.
- src_bytes: continuous.
- dst_bytes: continuous.
- land: symbolic.
- wrong_fragment: continuous.
- urgent: continuous.
- hot: continuous.
- num_failed_logins: continuous.
- logged_in: symbolic.
- num_compromised: continuous.
- root_shell: continuous.
- su_attempted: continuous.
- num_root: continuous.
- num_file_creations: continuous.
- num_shells: continuous.
- num_access_files: continuous.
- num_outbound_cmds: continuous.

- is_host_login: symbolic.
- is_guest_login: symbolic.
- count: continuous.
- srv_count: continuous.
- serror_rate: continuous.
- srv_serror_rate: continuous.
- rerror_rate: continuous.
- srv_rerror_rate: continuous.
- same_srv_rate: continuous.
- diff_srv_rate: continuous.
- srv_diff_host_rate: continuous.
- dst_host_count: continuous.
- dst_host_serror_rate: continuous.
- dst_host_srv_serror_rate: continuous.
- dst_host_rerror_rate: continuous.
- dst_host_srv_rerror_rate: continuous.

- num_access_files: continuous.
- class : symbolic

4 symbolic (nominal) + 39 numeric
Path to Gotcha Moment

- **Step 1:** Preprocessing
  - Normalization – Scaling
  - Class merging
  - Data Filtering

- **Step 2:** Feat. Selection
  - Genetic Algorithm (Out of curiosity)
  - Backward Selection
  - Supervision

- **Step 3:** Model Learning
  - Xval. Parameter optimizing
    - SVM
    - Neural Net.
    - Naive Bayes
    - Decision Trees

- **Step 4:** Rule Learning
  - Decision Trees
  - Naive Bayes

- **Step 5:** Eval. Of Anomaly detection
  - Get Useless Results by LOF (not included)
  - Compare with supervised method

Packages used:
- Rapidminer
- Knime
- Weka
- R

General Work Iteration
Run the code
Go out, waste time
Come back
Step1: Clear Data

• Remove redundant data
  – 4,898,431 -> 1,074,991 (742MB -> 18.7MB)
    • Attack : 3,925,650 -> 262,178 (cause deficiency ?)
    • Normal : 972,781 -> 812,814
      – Reduce majority effect

• Random Selection
  – 1,074,991 -> 125,973 (57,666 – Ano. Vs 67,388 - Normal)
    • Keep amount of fraud types same
    • Decrease payload of Normal instances (cost of Anomaly is bigger)

• Merge class values
  – FraudActivities (dos,probe) -> “anomaly”
Step 2: Future Selection

Genetic Algorithms vs Backward Selection + ROC curves Correlation Matrix
Step 2: Feature Selection

- Genetic algorithm:
  - NaiveBayes() { returns fitness; }

42 feats => 12 feats

**Without feature selection**

- Performance Vector:
  - Accuracy: **84.67%**
  - Confusion Matrix:
    - True: anomaly 10912, normal 1921
    - False: anomaly 1536, normal 8175

- Performance Vector:
  - Accuracy: **77.58%**
  - Confusion Matrix:
    - True: anomaly 8540, normal 4293
    - False: anomaly 761, normal 8950
Step 2

- **Backward elimination**
  - with Naive Bayes

  - PerformanceVector:
    - accuracy: **89.46%**
    - ConfusionMatrix:
      - True: anomaly 12257 normal 1800
      - anomaly: 8540 normal: 761

- **Without feature selection**
  - PerformanceVector:
    - accuracy: **77.58%**
    - ConfusionMatrix:
      - True: anomaly: 8540 normal: 761
      - anomaly: 4293 normal: 8950

**BETTER but LONGER!**

42 feats => 30 feats

**IMPROVEMENT!**

**GENETIC ALGO. Is good if you have time constraint!**

%84 Accuracy
Step 2: After unsupervised selection

42 features => 31 features

Selected features:
- protocol_type
- service
- flag
- land
- urgent
- hot
- num_compromised
- root_shell
- su_attempted
- num_root
- num_file_creations
- num_shells
- num_access_files
- num_outbound_cmds
- is_guest_login
- srv_count
- serror_rate
- srv_serror_rate
- rerror_rate
- srv_rerror_rate
- diff_srv_rate
- srv_diff_host_rate
- dst_host_count
- dst_host_same_srv_rate
- dst_host_diff_srv_rate
- dst_host_same_src_port_rate
- dst_host_srv_diff_host_rate
- dst_host_serror_rate
- dst_host_rerror_rate
- dst_host_srv_rerror_rate
Step2 with my hands

- Support of unsupervised selection
- What are my constraints
  - Correlation
    - Correlation = same information
  - ROC curves
ROC curves

Straight Lines = Uninformative
After ROC and Correlation

@ATTRIBUTE duration REAL
@ATTRIBUTE protocol_type REAL
@ATTRIBUTE service REAL
@ATTRIBUTE flag REAL
@ATTRIBUTE src_bytes REAL
@ATTRIBUTE dst_bytes REAL
@ATTRIBUTE land REAL
@ATTRIBUTE wrong_fragment REAL
@ATTRIBUTE urgent REAL
@ATTRIBUTE hot REAL
@ATTRIBUTE num_failed_logins REAL
@ATTRIBUTE logged_in REAL
@ATTRIBUTE num_compromised REAL
@ATTRIBUTE root_shell REAL
@ATTRIBUTE su_attempted REAL
@ATTRIBUTE num_file_creations REAL
@ATTRIBUTE num_shells REAL
@ATTRIBUTE num_access_files REAL
@ATTRIBUTE num_outbound_cmds REAL
@ATTRIBUTE is_host_login REAL
@ATTRIBUTE is_guest_login REAL
@ATTRIBUTE count REAL
@ATTRIBUTE srv_count REAL
@ATTRIBUTE rerror_rate REAL
@ATTRIBUTE same_srv_rate REAL
@ATTRIBUTE diff_srv_rate REAL
@ATTRIBUTE srv_diff_host_rate REAL
@ATTRIBUTE dst_host_count REAL
@ATTRIBUTE dst_host_diff_srv_rate REAL
@ATTRIBUTE dst_host_same_src_port_rate REAL
@ATTRIBUTE dst_host_srv_diff_host_rate REAL

Selected features

42 to 35 features
Naive Bayes after Supervised Filtering

--- Cross-Val Results ---
Correctly Classified Instances 117594 93.3486 %
Incorrectly Classified Instances 8379 6.6514 %
a b <-- classified as
64341 3002 | a = normal
5377 53253 | b = anomaly

--- Test Set Results ---
Correctly Classified Instances 16811 74.5697 %
Incorrectly Classified Instances 5733 25.4303 %
a b <-- classified as
9040 671 | a = normal
5062 7771 | b = anomaly

Clues of divergency of test datas!
Overfitting!
Final Attributes

42 to 29 features
Backward selec: -11
ROC+Corre. Mat: -2

Selected features

- protocol_type
- service
- flag
- land
- urgent
- hot
- num_compromised
- root_shell
- su_attempted
- num_root
- num_file_creations
- num_shells
- num_access_files
- num_outbound_cmds
- is_guest_login
- srv_count
- serror_rate
- srv_rerror_rate
- diff_srv_rate
- srv_diff_host_rate
- dst_host_count
- dst_host_same_srv_rate
- dst_host_diff_srv_rate
- dst_host_same_src_port_rate
- dst_host_srv_diff_host_rate
- dst_host_serror_rate
- dst_host_rerror_rate
- dst_host_srv_rerror_rate
Step 3: Model Learning

- Random Forests
- Neural Nets
- SVM with RBF
- Naive Bayes
Step3: Model Learning

• Random Forest
  – Enhanced to overfitting
  – No X-Val requirement

--- Summary ---
Correctly Classified Instances       18503       82.0751 %
Incorrectly Classified Instances      4041          17.9249 %

--- Confusion Matrix ---

a    b   <-- classified as
9022  689 |    a = normal
3352 9481 |    b = anomal

Step3: Model Learning

- Neural Nets
  - 10% of train data
  - (# classes+# attributes)/2 = # hidden units
  - Optimized Parameters
    - Momentum = 0.1
    - Learning rate = 0.2
    - Decay
  - Long, so long to train....

Accuracy: 84.08%
Step3: Model Learning

- **SVM – libSVM with RBF kernel**
  - Numeratize nominal values.
  - Long training time

Performance Vector:
- **accuracy**: 75.78%
- **Confusion Matrix**:
  
<table>
<thead>
<tr>
<th></th>
<th>anomaly</th>
<th>normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>anomaly</td>
<td>10342</td>
<td>2969</td>
</tr>
<tr>
<td>normal</td>
<td>2491</td>
<td>6742</td>
</tr>
</tbody>
</table>

Step3

• Naive Bayes
  – Probabilistic approach
  – Analitical results
  – Also gives probabilistic rules

GOTCHA!

PerformanceVector:
accuracy: 89.46%
ConfusionMatrix:
True:  anomaly normal
      anomaly: 12257  1800
      normal:  576  7911
Step4: Rule Induction

Decision Tree + Naive Bayes
Step 4: Backward Selection + Grid Search + Decision Tree

85.50% accuracy

=> 42 to 32 feats

<table>
<thead>
<tr>
<th></th>
<th>true anomaly</th>
<th>true normal</th>
<th>class precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>pred. anomaly</td>
<td>9866</td>
<td>303</td>
<td>97.02%</td>
</tr>
<tr>
<td>pred. normal</td>
<td>2967</td>
<td>9408</td>
<td>76.02%</td>
</tr>
<tr>
<td>class recall</td>
<td>76.88%</td>
<td>96.88%</td>
<td></td>
</tr>
</tbody>
</table>
Top Nodes

```
is_host_login = 0
| flag = OTH
| | protocol_type = tcp
| flag = REJ
| | protocol_type = tcp
| | | land = 0
| flag = RSTOS0
| | protocol_type = tcp
| | | land = 0
| flag = RSTOS0: anomaly {normal=0, anomaly=10}
| flag = RSTR
| | protocol_type = tcp
| | | land = 0
| flag = S0
| | protocol_type = tcp
| | | land = 0
| flag = S0
| | | logged_in = 0
| flag = S1
| | protocol_type = tcp
| | | land = 0
| | | | logged_in = 0
| | | | logged_in = 1: normal {normal=31, anomaly=0}
| flag = S2
| | protocol_type = tcp
| | | land = 0
| | | | is_guest_login = 0
| flag = S3
| | protocol_type = tcp
| | | land = 0
| | | | logged_in = 1
| flag = SF
| | land = 0
| | | protocol_type = icmp
| | | | logged_in = 0
| | | | protocol_type = tcp
| | | | hot > 1.500
| | | | hot ≤ 1.500
| | | | protocol_type = udp
| | | | logged_in = 0
| flag = SH: anomaly {normal=0, anomaly=28}
```

Most discriminative attributes

- **is_host_login**
  - ...at that particular snapshot

- **flag**
  - Package header flag

- **protocol_type**
  - Protocol of connection
Step 4: Naive Bayes

Accuracy 89.43%

Protocol Effect

Example Results

![Protocol Effect Chart]

- icmp
- tcp
- udp
- unknown

Density
Step 4: Naive Bayes

Accuracy 89.43%

# COMPROMISE CONDITIONS

- If error packages go up, chance of Fraud increases
Step 4: Naive Bayes

Accuracy 89.43%

FLAGS

- If so much Sync0 flag, increased prob of fraud
Gotchas

• Naive Bayes
  – Poor Machine + Less Effort + Naive Bayes = Best Model

• Best indicators of fraud
  – Protocols (monitoring icmp packages)
  – Log compromised conditions (larger # is secure)
  – Log SNY packages.
What could or will be done?

- Test selected features with other algorithms as well
- Craft a new algorithm for divergency
- Wire a software and see the real time performance.
- Run a association algorithm.
ANY QUESTION & COMMENT & SAYING & WORD & SOMETHING?