Data Mining Based Detection Methods

Feng Pan

Outline

- Related Data Mining Background
- Data Mining in Intrusion detection

Outline

- Related Data Mining Background
  - Pattern Mining
    - Association Pattern
    - Sequence Pattern
  - Association Rules
  - Classification
    - Decision Tree
    - CBA: Classification based on association rules

Typical Dataset in Data Mining

- Dataset consists of records.
- Records consist of a set of items.

r1	host_IP="10.11.0.1", host_port>1100, flag="SF"
r2	flag="SF"
r3	host_IP="10.11.0.1", host_port>1100, duration>100ms, bytes>1KB
r4	flag="SF",service="telnet", bytes>1KB
r5	host_IP="10.11.0.1", flag="SF", duration>100ms, bytes>1KB
r6	host_IP="10.11.0.1", host_port>1100, flag="SF"

Pattern Mining

- Association Patterns
  A set of items frequently occur, order doesn't matter.

Pattern1: host_IP="10.11.0.1", host_port>1100,
Pattern 2: flag="SF", bytes>1KB
Pattern 3: duration>100ms, bytes>1KB

Association Rules

- From the association patterns, we can get association rules (LHS->RHS)
  - cf is an association pattern, then we get the rule as flag="SF" -> bytes>1KB,
- Two measurements for the goodness of rules
  - Support: number of records containing ($LHS \cup RHS$) *support* (flag="SF" -> bytes>1KB)=2,
  - *Confidence:* number of records containing ($LHS \cup RHS$) / number of records containing (LHS) *confidence* (flag="SF" -> bytes>1KB)=40%

Classification

- Many different classifications
  - Decision Tree
  - Neural Network
  - CBA (Classification based on Association Rules)
- CBA is constructed based on the association rules.

CBA

r1	host_IP="10.11.0.1", host_port>1100, flag="SF"	normal
r2	flag="SF"	normal
r3	host_IP="10.11.0.1", host_port>1100, duration>100ms, bytes>1KB	abnormal
r4	flag="SF",service="telnet", bytes>1KB	normal
r5	host_IP="10.11.0.1", flag="SF", duration>100ms, bytes>1KB	abnormal
r6	host_IP="10.11.0.1", host_port>1100, flag="SF"	abnormal

•Add a new Column: Class Label

•Records are labeled by user using domain knowledge.

•Class Label can be considered as a special feature.

•We find association rule
duration>100ms, bytes>1KB ->abnormal
support=2, *confidence*=100%

•Then a simple rule of the classifier is
duration>100ms, bytes>1KB ->abnormal

CBA

r1	host_IP="10.11.0.1", host_port>1100, flag="SF"	normal
r2	flag="SF"	normal
r3	host_IP="10.11.0.1", host_port>1100, duration>100ms, bytes>1KB	abnormal
r4	flag="SF",service="telnet", bytes>1KB	normal
r5	host_IP="10.11.0.1", flag="SF", duration>100ms, bytes>1KB	abnormal
r6	host_IP="10.11.0.1", host_port>1100, flag="SF"	abnormal

•We can find many rules with different *support* and *confidence*
duration>100ms, bytes>1KB ->abnormal
:*support*=2,*confidence*=100%
host_IP="10.11.0.1", host_port>1100 ->normal
:*support*=2,*confidence*=66%
flag="SF",service="telnet"->normal
:*support*=1,*confidence*=100%

•Sorting all the rules according to their confidence and support.
1) duration>100ms, bytes>1KB ->abnormal
2) flag="SF",service="telnet"->normal
3) host_IP="10.11.0.1", host_port>1100 ->normal

CBA

- Given an unknown record, apply the rules in order.
  - "host_IP="10.11.0.1", host_port>1100, flag="SF"": apply rule 3 -> classify as class abnormal
  - "host_IP="10.11.0.1", host_port>1100, flag="SF", service="telnet"": apply rule2 -> classify as class normal
    - rule 3 can also apply to it, but rule2 has higher support and confidence
  - "host_port>1100, service="telnet", duration>100ms" : no rule can apply, then classify to default class
    - Random
    - Majority class

Outline

- Why Data Ming
- Challenge for Data Mining in intrusion detection.
- Two layers to use Data Mining
  - Mining in the connection data
  - Mining in the alarm records.

Why Data Mining?

- The dataset is large.
- Constructing IDS manually is expensive and slow.
- Update is frequent since new intrusion occurs frequently.

Can Data Mining work?

- Challenges for Data Mining in building IDS
  - Develop techniques to automate the processing of knowledge-intensive feature selection.
  - Customize the general algorithm to incorporate domain knowledge so only relevant patterns are reported
  - Compute detection models that are accurate and efficient in run-time

Challenge in feature selection

- Many features in the connection records, relevant or irrelevant.
- Automatic detection (classifiers) are sensitive to features. Missing of key features for some attack may result worst performance
  - The missing of "host_count" feature will make the IDS unable to detect DOS attack in the experiments on DARPAR data.
- Different attacks require different features
- Some useful features are not in the original data

Challenge in Pattern Mining

- Large amount of patterns can be found in the dataset. System may be overwhelmed.
- For different attacks, pattern mining shall focus on different feature subsets.
- For sequence patterns, different attacks has different optimal window size.

Challenge in Building Models

- Single model is not able to capture all type of attacks.
- An ideal model consists of several light weighted models each of which focuses on its own aspects.

Mining in the data

- Tow kinds of datset.
  - Network based dataset
  - Host based dataset
- Build IDS by mining in the records.
- When find attacks, give alarms to administration system.

Framework of Building IDS

- Step1: Preprocessing. Summarize the raw data.
- Step2: Association Rule Mining.
- Step3: Find sequence patterns (Frequent Episodes) based on the association rules.
- Step4: Construct new features based on the sequence patterns.
- Step5: Construct Classifiers on different set of features

Preprocessing

- To summarize raw data to high level event, e.g. a network connection (network based data) or host session (host based data).
- Bro and NFR can be used to do the summarizing.
  - Bro policy script
    - const ftp_guest_ids = { "anonymous", "ftp", "guest", } &redef;
      redef ftp_guest_ids += { "visitor", "student" };
      redef ftp_guest_ids -= "ftp";
  - NFR N-Codes

Association Rule Mining

- Customizing: Only report important and relevant patterns.
  - Define relevant features, reference features.
- Pattern must contain relevant features or reference features. Otherwise, the patterns are not interesting.

Association Rule Mining

- Example on the Shell Command Data.
  Shell Command Data is a host based dataset

Association rule	Meaning
$command = vi \rightarrow time = am$, $hostname = pascal$, $arg1 = tex$, [1.0, 0.28]	When using vi to edit a file, the user is always (i.e. 100% of the time) editing a *tex* file, in the morning, and at host *pascal*; and 28% of the command data has this pattern.
$command = subject \rightarrow time = am$, $hostname = pascal$, $arg1 = progress$, [1.0, 0.11]	The subject of the user's email is always ((i.e. 100% of the time) about "progress", in the morning, and at host *pascal*; and 11% of the command data has this pattern.

Table 4. Example Association Rules from Shell Command Data

Sequence Pattern Mining

- Frequent Episodes.
  - X,Y->Z, [*c*,*s*,*w*]
    - With the existence of itemset X and Y, Z will occur in time *w*.
    - example

Frequent episode	Meaning
(service = http, flag = S0, dst_host = victim), (service = http, flag = S0, dst_host = victim) → (service = http, flag = S0, dst_host = victim) [0.93, 0.03, 2]	93% of the time, after two *http* connections with S0 flag are made to host victim, within 2 seconds from the first of these two, the third similar connection is made, and this pattern occurs in 3% of the data

Example Intrusion Pattern

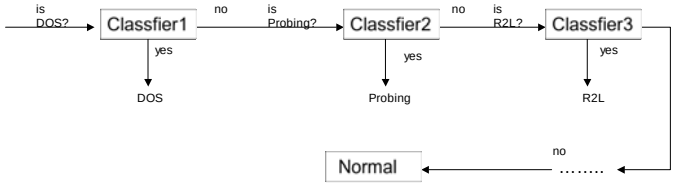
  - Different window size may generate different results.
  - Window size is related to attack type.
    - DOS: w=2 sec
    - slow Probing: w=1 min

Feature Construction

- Construct new feature according to the frequent episode.
  - Some features will show close relationship to each other. Then combine the features.
  - Some frequent episode may indicate interesting new features.

Build Model (classifier)

- Build different classifiers for different attacks.



The DARPA data

- 4G compressed *tcpdump* data of 7 weeks of network traffics.
- Contains 4 main categories of attacks
  - *DOS*: denial of service, e.g., ping-of-death, syn flood
  - *R2L*: unauthorized access from a remote machine, e.g., guessing password
  - *U2R*: unauthorized access to local super user privileges by a local unprivileged user, e.g., buffer overflow
  - *PROBING*: e.g., port-scan, ping-sweep

Preprocessing

- Use Bro script to summarize the raw data to records for each connection.
- Each connection contains some "intrinsic" features.
  - *time*, *duration*, *service*, *src_host*, *dst_host*, *src_port*, *wrong_fragment*, *flag*
  - *wrong_fragment*: e.g., fragment size is not multiple of 8 bytes, fragment offset are overlapped
  - *flag*: how the connection is established and terminated

Build training data

- Normal data set:
  - randomly extract sequences of normal connections records
- Data set for each attach type:
  - extract all the records that fall within a surrounding time window of plus and minus 5 minutes of the whole duration of each attack

Feature Construction

- Time-based "traffic" features: can detect *DOS* and *PROBING* attacks.
  - example

satan :- host_REJ_% $\geq$ 83%, host_diff_srv_% $\geq$ 87%.	If for the connections in the past 2 seconds that have same the destination host as the current connection, the percentage of rejected connections are at least 83%, and the percentage of different services is at least 87%, then this is a satan attack (a PROBING attack).

- "same host"
  - Exam only the connections in the past 2 seconds that have the same *dst_host* as the current one
  - ***Features:***
    count, *percentage of same service*, *percentage of different service*, *percentage of S0 flag*, *percentage of rejected connection flag*.

Feature Construction

- "same service"
  - Exam only the connections in the past 2 seconds that have the same *service* as the current one
  - ***Features****:*
    count, *percentage of different dst_host*, *percentage of S0 flag*, *percentage of rejected connection flag*.

Feature Construction

- Some *slow probing* attack need a larger window size
- Host-based "traffic" features
  - Instead of the time window of 2 seconds, use a connection window of 100 connections.
  - Same set of features on the connection window.
  - Can detect *slow Probing*.

Feature Construction

- *R2L* and *U2R* normally involve in a single connection and are embedded in the data portion of the package.
- "content" features can indicate whether the connection is suspicious.
  - *Number of failed login*.
  - *Successfully logged in or not*
  - ….
  - Example

rule	meaning
guess :- failed_logins >= 5.	If number of failed logins is greater than 5, then this telnet connection is "guess", a guessing password attack.

Build Model (Classifier)

- Build Classifier on each set of features
  - Time-base "traffic" features + intrinsic features
  - Host-base "traffic" features + intrinsic features
  - "content" features + intrinsic features

Build Model

- Example for the time-based "traffic" model

label	service	flag	host_count	srv_count	host_REJ_%	host_diff_srv_%	duration	...
normal	ecr_i	SF	1	1	0	1	0	...
smurf	ecr_i	SF	350	350	0	0	0	...
satan	user-level	REJ	231	1	85%	89%	0	...
normal	http	SF	1	0	0	1	3	...
...	...	...	...	...	...	...	...	...

Example "Traffic" Connection Records

RIPPER rule	Meaning
smurf :- service=ecr_i, host_count ≥ 5, host_srv_count ≥ 5.	If the service is icmp echo request, and for the past 2 seconds, the number of connections that have the same destination host as the current one is at least 5, and the number of connections that have the same service as the current one is at least 5, then this is a smurf attack (a DOS attack).
satan :- host_REJ_% ≥ 83%, host_diff_srv_% ≥ 87%.	If for the connections in the past 2 seconds that have same the destination host as the current connection, the percentage of rejected connections are at least 83%, and the percentage of different services is at least 87%, then this is a satan attack (a PROBING attack).

Rules for DOS and PROBING attacks

Build Model

- Example for "content" model

label	service	flag	hot	failed_logins	compromised	root_shell	su	duration	...
normal	telnet	SF	0	0	0	0	0	10.2	...
normal	telnet	SF	0	0	0	3	1	2.1	...
guess	telnet	SF	0	6	0	0	0	26.2	...
normal	telnet	SF	0	0	0	0	0	126.2	...
overflow	telnet	SF	3	0	2	1	0	92.5	...
normal	telnet	SF	0	0	0	0	0	2.1	...
guess	telnet	SF	0	5	0	0	0	13.9	...
overflow	telnet	SF	3	0	2	1	0	92.5	...
normal	telnet	SF	0	0	0	0	0	1248	...
...	...	...	...	...	...	...	...	...	...

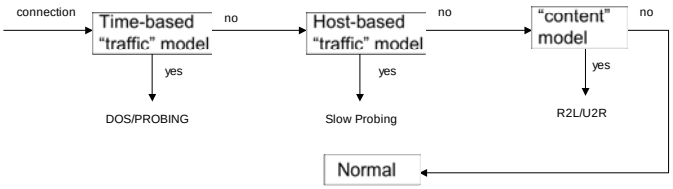
Telnet Records

RIPPER rule	Meaning
guess :- failed_logins >= 5.	If number of failed logins is greater than 5, then this telnet connection is "guess", a guessing password attack.
overflow :- hot = 3, compromised = 2, root_shell = 1.	If the number of hot indicators is 3, the number of compromised conditions is 2, and a root shell is obtained, then this telnet connection is a buffer overflow attack.
...	...
normal :- true.	If none of the above, then this connection is "normal".

Rules from Telnet Records

Build Meta-Classifier

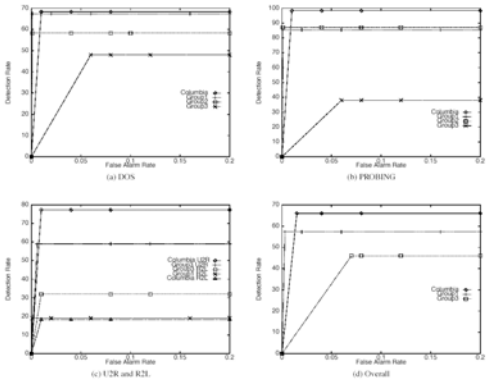
- Build Meta-Classifier to combine all the 3 classifiers.



Results

- Training on the 7 weeks of labeled data, and testing on the 2 weeks unlabeled data.
- The test data contains 14 attack types which do not exist in training data.
- Comparing 4 methods:
  - Columbia: the IDS developed according to the framework introduced above
  - Group 1-3: three systems developed by knowledge engineering approaches.

Results



Results

- Detection rate on New and Old attacks.
  - Old attacks: type of attacks occur in both training and testing data.
  - New attacks: type of attacks occur in testing data only.

Category	Old	New
DOS	79.9	24.3
PROBING	97.0	96.7
U2R	75.0	81.8
R2L	60.0	5.9
Overall	80.2	37.7

Table 10. Comparing Detection Rates (in %) on Old and New Attacks

Mining in the alarm records

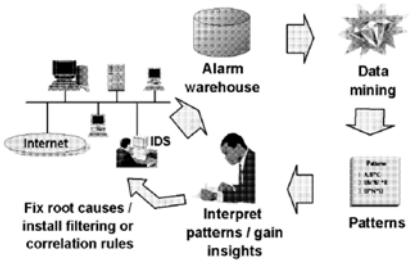
- IDS may generate 100,000 alarms every month.
  - Example:

	alarm source	alarm destination	alarm type
r1	10.11.0.1	102.10.0.2	TCP Fin Host Sweep
r2	10.11.0.2	102.10.0.2	Orphaned Fin Packet

  - Some alarms are false positive.
  - Some alarms are redundant, closely related to some others.
- The analysis on alarms can help IDS administrators to refine the rules, set filters to get rid of redundant alarms.

Framework of Alarm Investigation

- Iteration process.



Methods

- To analyze the alarm records and make refinement, 2 possible methods can be used.
  - Frequent Episode
    Find Frequent Episodes in the alarm records.
  - Attribute-Oriented Induction
    Cluster the alarms into groups

Frequent Episodes

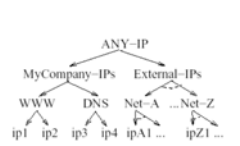
- Frequent sequence mining in the alarm records.
- Sequences can show the relationship between different alarms
  - alarm1 happens -> alarm2 happen in 2 sec.
    Refine rules to alert 2 in advance when alarm 1 occurs.
  - alarm 1 and alarm2 always happens together
    one of the alarm is redundant.

Frequent Episodes

- Frequent Episodes does not work well on alarm mining
  - Takes a long time to do mining in the large amount of alarm records.
  - Discovered frequent episodes can only cover a small portion of the alarm records. Large amount of records are still remained for manually analyzing.

Attribute- Oriented Induction (AOI)

- Framework
  - Step1: select one feature, for each records, generalize value of that feature.
  - Step2: combine records that are identical on all feature values.
  - Step3: repeat the above two steps until data are generalized enough.
- Basically, it is a clustering process.
  - Attribute values are maintained in a family structure.



a) Generalization hierarchy for IP addresses.

SrcIP	DstIP	Count
ip1	ip4	1000
ip1	ipA1	1
ip1	ipB1	1
⋮	⋮	⋮
ip1	ipZ1	1
ipA1	ip4	1
ipB1	ip4	1
⋮	⋮	⋮
ipZ1	ip4	1

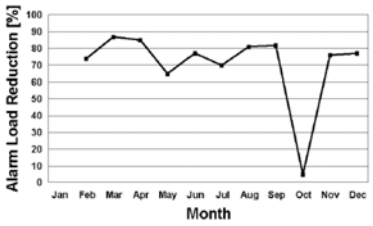
b) Sample table.

The classical AOI algorithm

```
1: for all alarms a in T do a.C := 1; // Init counts
2: while table T is not abstract enough do {
3:       Select an alarm attribute A_i;
4:       for all alarms a in T do          // Generalize A_i
5:            a.A_i := father of a.A_i in H_i;
6:       while identical alarms a, a' exist do    // Merge
7:            Set a.C := a.C + a'.C and delete a' from T;
8: }
```

Experiment Results

- The chart below shows the Alarm load reduction on one of the IDS tested. For the month of Oct., the reduction is low because of a networking problem in that month.



Alarm load reduction for IDS 8.

Conclusion

- Data mining techniques are very useful in Intrusion Detection
- Still need manually interpretation/advice in some processing steps
- More efficient on known attacks than on unknown attacks.
  - Only if the training data contains all normal behavior