|  |
| --- |
| Data Analysis in enterprise security |
| Fingerprint enterprise user behavior |
| Enhong Ma |

Fingerprint enterprise user behavior

Contents

[1) Introduction 1](#_Toc416018484)

[2) Current Challenge 1](#_Toc416018485)

[3) Scope and Description of Work 2](#_Toc416018486)

[4) Theory and Methods 2](#_Toc416018487)

[5) Gather the evidence 4](#_Toc416018488)

[6) Normalize data 5](#_Toc416018489)

[7) Reduce type 1 error 6](#_Toc416018490)

[8) Communicate to data 7](#_Toc416018491)

[9) Developing security-driven benchmarks 8](#_Toc416018492)

[10) Privacy 8](#_Toc416018493)

[References: 9](#_Toc416018494)

# Introduction

The biggest threat to organizations is their lack of visibility into their networks and the amount of time it takes them to find the threat actors. Most organizations don't have a strategy for dealing with this identification problem once the attacker gets inside -- an event that is almost inevitable. How can company quickly differentiate normal behavior from abnormal actions among hundreds of thousands enterprise events in real time? User behavior fingerprint can be used as a strategy for dealing with this problem.

# Current Challenge

Traditional security mechanisms have worked by using rule, pattern, signature and algorithm-based approaches to detect malicious software or suspicious activity. These approaches require constant care and feeding to identify and mitigate security threats. The problem is that those rules are built on events that happened in the past. It’s based on what you know, not what you don’t know. Especially, it is lacking the ability to correlate events to create a “dynamic” pattern. Snowden did not just arrive the first day and started stealing information. Hundreds Giga data from company network of Sony Pictures Entertainment can’t be transferred out in one day. Virus rooted in Target retail pose machine will take several days to find its way to internal network and open reversal tunnel to hack sites. All these events are deeply buried in unstructured enterprise activity log. If we could digitalize user behavior by fingerprinting their usual activities, as the software becomes familiar with the ways in which a particular person or system functions, it can then detect anomalous behavior. It’s similar to the ways that credit card companies have used technology to learn when and where customers typically use their cards to identify suspicious purchases.

With faster identification of vulnerabilities, increased detection rates and the discovery of attack vectors that were previously undetectable, we would be alerted once abnormal events happened.

# Scope and Description of Work

We first must build up the base. It should include necessary business behavior, regular user usage pattern and normal events. We then should consider how to collect the data and analyze correlation between them.

After that, we should monitor above determined events and create digital fingerprint that reflect how a person operates and how regular business events should behave.

# Theory and Methods

Currently, the notion of secure learning largely remains ill-defined and domain-specific and lacks proper evaluation. There is a need to incorporate a realistic model of the adversaries, to formalize secure learning, and to identify how enterprises can benefit from these technologies.

We decide to use machine learning to create the digital fingerprint automatically. Machine-learning methods can be categorized into four groups of learning activities:

1. symbol-based
2. connectionist-based
3. behavior-based
4. immune system-based

Symbol-based machine learning has a hypothesis that all knowledge can be represented in symbols and that machine learning can create new symbols and new knowledge, based on the known symbols. In symbol-based machine learning, decisions are deducted using logical inference procedures. Connectionist-based machine learning is constructed by imitating neuron net connection systems in the brain. In connectionist machine learning, decisions are made after the systems are trained and patterns are recognized.

Behavior-based learning has the assumption that there are solutions to behavior identification, and is designed to find the best solution to solve the problem.

The immune-system-based approach learns from its encounters with foreign objects and develops the ability to identify patterns in data.

Since enterprise events will include system usage, network activity and user behavior pattern. We will implement hybrid learning. Here are security events we are planning to collect to profile the fingerprints.

|  |  |  |
| --- | --- | --- |
| Type of events | Input data | Methods used |
| hosts | frequency of system calls; number of system calls | Classification and linear regression trees |
| User | total active time; connection time window; network access pattern and volume | Classification; clustering; correlation |
| network | territory of access destination; throughput; | Unsupervised clustering algorithm; correlation; decision tree |

We use rule to describe the correlation between attribute conditions. These rules can be used to generate user profile fingerprint. If user’s activities are found not to be consistent with the established rules, then we consider it potential security threat. We will uses association rules to capture and represent above events relationship. Association rules classification describes the frequent patterns in a multidimensional data set. Association rules are generated in two steps. First, we find all frequent item sets and identify the strong association rules in the frequent item sets. Mining frequent item sets from a large data set is challenging, because it generates a large number of item sets, which satisfy the minimum support threshold, and if any item set is frequent, its subset should also be frequent. Researchers have proposed many efficient algorithms for association rule mining. Among these methods, an apriori algorithm introduced by Agrawal et al. in 1993 is the most commonly used frequent association rule mining algorithm. This algorithm uses support and confidence measures of interestingness and improves rule mining efficiency by using the prior knowledge of frequent item set properties that all nonempty subsets of a frequent item set must also be frequent. Here is a host based record data.

|  |  |  |  |
| --- | --- | --- | --- |
| Time | Hostname | Command | Arg |
| 9-12pm | Desktop1 | Cd | home |
| 9-12pm | Desktop1 | Vi | \*.py |
| 9-12pm | Desktop1 | Email | manager |
| 9-12pm | Desktop1 | Vi | \*.py |
| 1pm – 5pm | Desktop1 | Chrome | Bbc.com |
| 1pm – 5pm | Desktop1 | Mplayer | Music files |
| 1pm – 5pm | Desktop1 | Vi | \*.py |

After apply the association rules, we will get followings:

|  |  |
| --- | --- |
| Association rules | Meaning |
| Command = vi => time =am | User uses vi to do python development in the morning. Most emails are to manager to report progress |
| Host = Desktop1 |
| Command = chrome => time = pm  Command = mplayer => time = pm | User browse new and listen music in the afternoon |
| Command = vi => time =pm | Also do development in the afternoon |

With the this result, each user can be represented with a multidimensional array including time, IP, frequent used commands, servers frequently being used, remote access web site, traffic type etc. These are all the factors including in this user’s fingerprint. Once user fingerprint is generated, we can proceed next step to normalize user activity boundary.

We will adapt both supervised and unsupervised algorithms in this stage. The objective of using a supervised algorithm is to obtain the highest classification accuracy. The most popular supervised machine-learning methods include artificial neural network (ANN), support vector machine (SVM), decision trees, Bayesian networks (BNs), k-nearest neighbor (KNN), and the hidden Markov model (HMM). Here we choose to use k-nearest neighbor approach to determine the similarity between users in the same working group. It is under assumption that people in the same department, their fingerprint should be close to each other in KNN.

In order to use machine learning to try and stop high-risk employee behavior and insider threats, we are about analyzing changes in behavior and predicting risks and breaches before they happen. All these behaviors don’t have a pre-defined pattern and we don’t have training and testing data as supervised learning needs. For example, in order to identify people who are about to leave the company with company data, we might see longer working day, his network throughput is increasing, he accessed file server more often and he has lots of data transfer traffic. With all these abnormal out of boundary events, machine learning can detect changes in user behavior that may suggest an employee is getting ready to leave the company with company data or it might alert for further investigation. For these novel events, we consider to implement the cluster analysis. In general, there are two purposes for using cluster analysis: understanding and utility. Clustering for understanding is to employ cluster analysis for automatically finding conceptually meaningful groups of objects that share common characteristics. It plays an important role in helping people to analyze, describe and utilize the valuable information hidden in the groups. Clustering for utility attempts to abstract the prototypes or the representative objects from individual objects in the same clusters. These prototypes/objects then serve as the basis of a number of data processing techniques such as summarization, compression, and nearest-neighbor finding.

Many network related events data are in binary stream, which means they are rapid and continuous arrival online, need for rapid response, potential boundless volume. Domingos and Hulten (2001) employed the Hoeffding inequality [[a]](#paper1) for the modification of Kmeans clustering, and obtained approximate cluster centroids in data streams, with a probability-guaranteed error bound [[b]](#paper2). Ordonez (2003) proposed three algorithms: online K-means, scalable K-means, and incremental K-means, for binary data stream clustering [[c]](#paper3). These algorithms use several sufficient statistics and carefully manipulate the computation of the sparse matrix to improve the clustering quality. Experimental results indicate that the incremental K-means algorithm performs the best.

The unsupervised machine-learning methods we will use is k-means clustering. Clustering is the assignment of objects into groups (called clusters) so that objects from the same cluster are more similar to each other than objects from different clusters. The sameness of the objects is usually determined by the distance between the objects over multiple dimensions of the data set. k-Means clustering partitions the given data points X into k clusters, in which each data point is more similar to its cluster centroid than to the other cluster centroids.③

The k-means clustering algorithm generally consists of the steps described as follows:

Step 1. Select the k initial cluster centroids, c1, c2, c3 , ck.

Step 2. Assign each instance x in S to the cluster that has a centroid nearest to x.

Step 3. Recompute each clusters centroid based on which elements are contained in it.

Step 4. Repeat Steps 2 through 3 until convergence is achieved.

Once user events are represented in multidimensional array, we can use Euclidean to form many events similarity clusters. These clusters can be considered to be the base of enterprise events.

# Gather the evidence

We first plan how and where to collect the original data and prepare data for analysis.

The volume of data is extremely large, and it requires data reduction in data preprocessing. Additionally, most of the data in the network are streaming data, and requires further data reduction. Thus, the data preprocessing step includes feature selection, feature extraction, or a dimensionality reduction technique, and an information-theoretic method.

Effective information monitoring builds on data collected from multiple sensors that

generate different kinds of data and are created by many different people for many

different purposes. A sensor can be anything from a network tap to a firewall log; Building up a useful sensor system requires balancing its completeness and its redundancy. A perfect sensor system would be complete while being nonredundant: complete in the sense that every event is meaningfully described, and nonredundant in that the sensors don’t replicate information about events.

No single type of sensor can do everything. Network-based sensors provide extensive coverage but can be deceived by traffic engineering, can’t describe encrypted traffic, and can only approximate the activity at a host. Host-based sensors provide more extensive and accurate information for phenomena they’re instrumented to describe. Here are several common log formats in network analysis, including HTTP ELF and CLF, SMTP log messages, Microsoft exchange message tracking logs(MTL),syslog.

Data storage is the basic problem in information security analysis as security events are scattered in a vast number of innocuous logfiles in different format. Data must be available for rapid data access. The major choices are flat files, traditional RMDS database and NoSQL DB. Here is comparison of three methods

|  |  |  |
| --- | --- | --- |
| Type of storage | Pros | Cons |
| Flat file | Easy access, simple to read and analyze; simple parsing tool; | Lack particular tools for optimized access; inefficient storage |
| RMDS | Mature solutions; well-defined interface language; stable and scalable solution | Not designed to deal with log data; data must be normalized before stored in DB. |
| NoSQL | Extremely powerful and well distributed query tools. Designed to handle unstructured data like log file, document and event record | Require programming and system administration skills as well as significant hardware commitment. |

# Normalize data

Majority data we collected are through system log, audit log and event log. These files are in a format that is designed mainly for consumption by human eyeballs. We would like computers to consume these data and ideally remove invalid “bad” data from data set. To normalize data, we need to coordinate with different business stake holder to build up good domain knowledge to understand the inputs. Spending enough time to prepare the data sometime is more important than building up a good analysis model. At the same time, we have to realize that there is error in collected data. Here is basic strategy before reporting error data:

* 1. Is reporting error in your data likely to not be random? If so, your results will be biased.
  2. Do you think reporting error is more likely in some variables than in others?
  3. Can you use other variables to inform sources of potential bias?
  4. Does the structure of the data itself change the probability of reporting error?

# Reduce type 1 error

When system decides which event should trigger alert, it uses hypothesis testing. It is the process of evaluating an out of boundary event. This kind of event might be that the data is normally distributed, or security issue. Here is our hypothesis:

1. Null hypothesis(Ho) – it is normal event and no action is needed
2. Alternative hypothesis(Ha) – potential security event and attention is needed.

Where a positive result corresponds to rejecting the null hypothesis(consider alternative hypothesis, mean there is security risk), and a negative result corresponds to not rejecting the null hypothesis(means there is no security risk) . During the evaluation, we call false positive type 1 error, false negative type 2 error. Although type 2 error is more harmful than type 1 error, we must control type 1 error rate to lower the false alert rate. In our above behavior based analysis, one of the downside is high type 1 error since user events are dynamic changing and most changes are not security concerns. In order to determine if an event is significant, we need to determine the probability of the result happening by chance. In statistical testing, this is done by using a p-value. The p-value is the probability that if the null hypothesis is true, you will get a result at least as extreme as the observed results. The lower the p-value, the lower the probability that the observed result could have occurred under the null hypothesis. Conventionally, a null hypothesis is rejected when the p-value is below 0.05. In our system, we can apply below integration methods to reduce the false alarm.

Real alert

Safe event

Events fingerprint profile generator

No

compare

Alarm?

Misuse detection or manual interfere

Yes

security event

Baseline of K-means cluster

If it is false alert, send feedback to update baseline cluster

Before deciding it is a real security event, we filter them through a rule based misuse detection layer.

The misuse detection methods for detecting malicious code use rule-based policy to identify threats in real-time, using adaptive threat detection with temporal data modeling and missing data. One of the limitations underlining most of the rule-based approaches is that they treat each event as a separate activity without considering the context of the events and can’t deal with novel attack, which doesn’t have an existing rule defined. When we attach this after the behavior analysis system, we have the opportunity to apply more statistical algorithm to reduce false alarm. As we talked before, rule-based approach has the limitation that it treat each event as a separate activity, we can perform inference between rules using Bayesian statistics(BN) to specify the causal relationships between subsets of variables.

The naïve Bayes (NB) classifier makes the assumption of class conditional independence. Given a data sample, its features are assumed conditionally independent of each other. This assumption is different from the assumption in BN that dependencies exist between features. In this sense, NB is a special and simple case of BN. In Ankita K Tiwari (2013)[[d](#paper4)], naïve Bayes was used to detect new, previously unseen malicious executables accurately and automatically. The method was compared with a traditional signature-based method, and it more than doubled the detection rates for new malicious executables.

# Communicate to data

After collecting/analyze the data, we need to transform security to more specific and measurable as an approach to inform the decisions or actions in context. I would like to suggest dashboard with interactive visualizations.

There is an inherent “call to action” nature to dashboards, with each element being either quantitative (has a value) or categorical (a list of items). Most of us have a great deal of readily available quantitative data related to information security ranging from lost assets, to security incidents, to SIEM events-per-second, to firewall/IPS operational data. In order for this data to be useful in the context of a dashboard, these quantitative measures must be able to answer two questions:

● What’s going on?

● So what?

For the categorical measures, you are usually identifying a set of elements that:

● Provide useful information—such as “which incident handlers are primary for the day?”;

● Require the most attention—such as “which Payment Card Industry Data Security Standard (PCI DSS) controls are slipping?”; or

● Need follow up—such as “what are the top expedited firewall port open requests?”

In order to help people to understand and learn from data, effective communication is the primary goal of visual/dashboard creations. Previous rule of developing simple and successful fixed dashboard probably can meet 95 percent of communication request. There are situations when static views are either insufficient or just not practical, requiring the move to a more dynamic and interactive medium. Here we will discuss some basic rules to help move from static to interactive.**①**

**Augmentation**—If adding interactive capabilities helps speed up or automate tasks consumers would normally perform manually, going interactive is definitely the right thing to do. There are many repetitive, time-consuming, data-driven tasks in information security. Logs must be collected and correlated, alerts must be received and attended to, and anomalies must be investigated. These actions often involve running a variety of utilities over individual pieces of data or sets of data elements to determine whether there truly is an issue on network. A research team led by Robert Erbacher worked to understand both the problem domain and how incident responders think and process information. This resulted in the creation o f VisAlert(http://digital.cs.usu.edu/~erbacher/publications/visalert.pdf).

**Exploration**—If the number of dimensions and size/diversity of the data set grow sufficiently large, it may be better to enable consumers to explore the relationships and outcomes on their own rather than trying to guess which set of static graphics will be most useful. There is an innovative open source tool released in 2013 by John Goodall called the Nessus Vulnerability Explorer (NV) (http://ornl-sava.github.io/nv/#). NV allows you to take an export from your Nessus scans, drag the file right into your browser, and begin exploring the vulnerabilities contained within.

**Illumination**—If a topic is complex enough, it may help to provide a well-executed, interactive visualization that provides a user-friendly interface for directed/constrained navigation around the data you’ve chosen to present.

# Developing security-driven benchmarks

As our system is taking feedback from the false alert and update the original fingerprint accordingly, we need a monitor system to check the integrity of our fingerprint data set. The purposes of this checking is ensure our base data set is not polluted by malicious data.

# Privacy

When profiling user activity and enterprise event, we must consider the data privacy. This includes collecting business event data and user private information reliably, store data securely, process data safely and disclosing result to proper group. All these stages must comply with government regulation. Here we will discuss the PPDM(privacy preserving data mining). Preserving privacy is nearly ubiquitous in various informatics disciplines, including but not limited to bioinformatics, homeland security, and financial analysis. It influences cybersecurity significantly with the recent development of information collection and dissemination technologies. The unlimited explosion of new information through the Internet and other media have inaugurated a new era of research where data-mining algorithms should be considered from the viewpoint of privacy preservation, called privacy-preserving data mining (PPDM). The Online Security and Privacy Study of 2009 conducted among 2385 U.S. adults showed a 78% increase from 2007 respondents who choose to log on Internet browsers that protect private information, and a 62% increase in respondents who choose servers that provide built-in security (Online Security and Privacy Study, 2009).

The ubiquitous data analysis algorithms allow malicious users to employ data mining to obtain private information and, hence, raises the following questions:

* will data mining compromise privacy?
* should data mining be limited in some applications?

This concern can be addressed from two aspects: ethical and technological. Legitimate use of private data would benefit the data-mining users and private owners. Many countries have produced regulations and legislation to protect the data owners, control the dissemination of private data, and regulate the accuracy of a database.

From technology perspective, we use PPDM to prevent unauthorized users from accessing private information, such as private data-mining or machine-learning results. Privacy preservation and data mining worked in parallel, until Aggrawal et al. defined the specific research area in data mining concerning privacy protection in 2000. In PPDM, researchers adopt a large number of privacy preservation techniques in data-mining and machine-learning algorithms to preserve knowledge security. The complexity in PPDM algorithms raises several research topics other than privacy preservation and data mining. Verykios et al. (2004a) classified the existing PPDM techniques by considering five views: **horizontal or vertical data distribution, data modification methods, data-mining algorithms, rule confusion, and privacy preservation.** Most data distributions are horizontal or vertical. We classify the PPDM methods into three groups according to data resource distribution: heuristic privacy preservation, cryptographic privacy preservation, and randomization/perturbation/reconstruction-based privacy preservation. The first group of techniques includes centralized data resources; the second group includes distributed computation, or SMC; and the third group includes both data resources. Centralized data-based PPDM refers to data-mining or machine-learning algorithms that perform applications on the data resources collected at a single central repository. The data storage system can be violated for privacy easily from the inference or learned rules. The proposed solutions employ perturbation-based or blocking-based rule confusion methods to downgrade the rule mining results so that sensitive rules are buried in insensitive rules or insensitive information blurs sensitive data. Heuristic privacy preservation is also called downgrading privacy preservation, because the data modification leads to the downgrading of rules or the ineffectiveness of machine-learning classifiers. Data-mining or machine-learning algorithms must balance between mining or learning efficiency and privacy preserving. Another solution to the centralized data system is to distribute data among multiple repositories and minimize the information leaks by applying SMC. SMC takes advantage of the proliferation of Internet technologies and cooperative computations on private data resources. Researchers have investigated various data-mining algorithms in isolation of each other. Among them, the most important privacy preservation methods have been proposed for a number of data-mining algorithms, like support vector machines (SVM) classification (Yu et al., 2006), association rule mining algorithms (Evfimievski et al., 2004), K-means clustering (Vaidya and Clifton, 2003), decision tree inducers (Agrawal and Srikant, 2000), BN (Wright and Yang, 2004), KNN (Kantarcioglu and Clifton, 2004), ANN (Barni et al., 2006), and other statistical methods (Du et al., 2004).

# References:

Research paper:

1. Bentkus, V.: On hoeffding’s inequalities. Ann. Probab. 32(2), 1650–1673 (2004)
2. Domingos, P., Hulten, G.: A general method for scaling up machine learning algorithms and its application to clustering. In: Proceedings of the 18th International Conference on, Machine Learning, pp. 106–113 (2001)
3. Ordonez, C.: Clustering binary data streams with k-means. In: Proceedings of the SIGMOD

Workshop on Research Issues in Data Mining and Knowledge Discovery (2003)

1. Classification of Malware based on Data mining approach

Books:

1. ***Interactive Data Visualization for the Web***
2. ***http://www.cert.org/flocon/2008/presentations/flocon08-mchugh-vagi.pdf***
3. ***Naked statistics: stripping the dread from data***
4. ***Data Smart: Using Data Science to Transform Information into Insight***
5. ***Data Mining and Machine Learning in Cybersecurity***
6. ***Tidy Data by Hadley Wickham***
7. ***Bad data handbook by Q. Ethan McCallum***
8. ***K-means Clustering – A Data Mining thinking***