Secure because Math: A deep-dive on Machine Learning-based Monitoring

Alex Pinto

alexcp@mlsecproject.org / @alexcpsec

Chief Data Scientist of MLSec Project

Black Hat Briefings USA 2014

Introduction and Abstract

We could all have predicted this with our magical Big Data analytics platforms, but it seems that machine learning is the new hotness in Information Security. A great number of start-ups with 'cy' and 'threat' in their names that claim that their product will defend or detect more effectively than their neighbors' product "because math." And it should be easy to fool people without a PhD or two that math just works.

Indeed, math is powerful and large-scale machine learning is an important cornerstone of much of the systems that we use today. However, not all algorithms and techniques are born equal. Machine learning is a very powerful toolbox, but not every tool can be applied to every problem and that's where the pitfalls lie.

This presentation will describe the different techniques available for data analysis and machine learning for information security, and discuss their strengths and caveats. The ghost of marketing past will also show how similar the unfulfilled promises of deterministic and exploratory analysis were, and how to avoid making the same mistakes again.

Topics

Motivation

It should come as no surprise to Information Security practitioners that there is an overwhelming amount of information being generated by operating systems, applications and network devices. Web services, routers, switches and anti-malware solutions are constantly generating logs and other information about their operation, just to name a few.

And this generates a lot of anxiety. A fear of missing out (FOMO) on all this information that is being generated on these devices.

On one hand, the fact that we are progressively more worried about the output of the tools means that in our mind we are starting to leave behind the idea that information security is all about "prevention" (i.e. automatic blocking), and the important concepts of "detection" and "response" are more and more into the limelight.

However, we are constantly being plagued with the same issue over and over again. A recent (May 2014) white paper from ESG "Reducing the Critical Time from Incident Detection to Containment" [1] again parrots that we are understaffed, doing everything manually and having too many diverse tools that do not speak to each other. What else is new, eh?

So I guess it should not come as any surprise that there is such a heavy push for Information Security Analytics from the vendors that, in a way, have put us into the situation we find ourselves right now.

I have also been responsible for trying to bring this discussion to the mainstream [2], so I should be happy that there are more people working and discussing this subject, right?

But I guess I could not predict (pun intended) the amount of interest that would come out of the idea, and the number of nascent companies engaging in the space. The fact is that we seem to be engaging on a familiar cycle of over-promising and under-delivering, which may be responsible for making a whole industry pass up the opportunity of adding the wonders of data analysis and machine learning to its fold.

As I have often heard that one of the biggest problems of Information Security selling is the asymmetric amount of information that the sellers hold, I'd like to cover some of the most important points on applying analytics and data mining to our field, based on historical facts and extensively exploitative marketing materials available.

This white paper assumes that the reader has some familiarity with Machine Learning terms and types of algorithms. If you have some doubts on the difference between supervised and unsupervised learning, please review my white paper from last year [2].

Brief History of Information Security Analytics and Automation

I find it interesting that most of the companies that are riding the analytics wave in Information Security right now are brand new startups. It would seem that having a trendy name with "cy" or "threat" (or more recently, "ray" or "beam") and crying "big data analytics" would be what it takes to get funded these days.

But what amazes me is that none of these ideas and implementations is novel. In fact, with just a little digging on academic research in Information Security we can see attempts (and mostly failures) to implement those same concepts that are getting people crazy money.

It is an interesting exercise to compare the way that behavioral analysis is explained and advertised for the last 20 years:

- "The statistical subsystem maintains historical profiles of usage per user and raises an alarm when observed activity departs from established patterns of usage for an individual." -- NIDES (1993)
- "Mathematical models can be developed that determine baseline behavior across users and machines, detecting (...) anomalous and risky activities (...)" - FileTrek (2014)
- "(...) monitors employee deviations from standard behavior (like if they suddenly access hundreds of files)."-- Adallom (2014).

Data Analysis in Network Monitoring

The first IDS on record was based on research by Dorothy E. Denning in SRI International (Stanford Research Institute) in 1986! The solution, called Intrusion Detection Expert System had a dual approach with a rule-based Expert System to detect known types of intrusions plus a statistical anomaly detection component based on profiles of users, host systems, and target systems.

Also in 1993, the same group released a new version called Next-Generation Intrusion-Detection Expert System [3]. And you would think we are being innovative calling our newer versions Next Generation.

As some of you will remember in 1998, Bro was created, followed by Snort in 1999. Those were the years that the concept of using anomaly detection in information security monitoring hit the mainstream with the DARPA Intrusion Detection Evaluation [4] datasets that were released in 1998 and 1999 in conjunction with the MIT. These datasets consisted of TCPdumps and Solaris BSM audit data [5] and were widely used as the source of research for anomaly detection, with over 5050 citations or references in Google Scholar [6].

On the same year of 1999 there was another publicly released network intrusion dataset, the KDD (Knowledge Discovery and Data Mining) Cup 1999 dataset [7], which would become even more popular, with over 6200 citations in Google Scholar [8]

And there lies the problem. From the beginning of this year alone, more then 300 papers are citing and using these datasets for anomaly detection research purposes. Because obviously the nature of the threats we face have not evolved in the past 15 years. An interesting associated fact is that there is a paper from

2000 by McHugh from that shows that the DARPA datasets are not appropriate to simulate actual network systems [9].

It was this research effort that made the first commercially available network anomaly detection as early as 2004, with the ISS Network Anomaly Detection product (which has been rolled up with their QRadar offering).

Ever since, the same anomaly detection DNA has been present and further developed into countless Netflow-based anomaly detection products, which have dealt with various degrees of success. A few that have shown some promise have been acquired by some of the larger players in the industry.

Data Analysis in SIEM and Log Management

The same concepts have also been extensively applied in log management and SIEM solutions, as far as the early 2000s as well. A lot of the work that has been poured into this has been around basic statistical analysis and the creation of association rules:

- Examples of basic statistical analysis are usually around the creation of baselines and trends into specific fields or rules, again, very open to interpretation by the analysts or the rule designers
- The mining and creation of association rules has been observed in ArcSight as soon as 2004 in their pattern discovery offering in version 3.0 of their product and is now being replicated by SumoLogic (not surprisingly, since the founders there were the ones who developed it on ArcSight in the first place)

Also, the analysts themselves have been trying to do this ever since they first started to get their hands in the data. There are some good documentation of sample scripts and analysis on those early days that can be used for examples [10].

But there is a long way to go. There are some issues that make anomaly detection especially hard to be applied into the information security field, and we discuss them further into this document.

Machine Learning and Data Analysis Capabilities in Information Security

Anomaly Detection

Anomaly detection is often used when the researchers are not really sure what they are looking for, based on an assumption of underlying normalcy. It this underlying structure or pattern is affected, then there must be something "bad" going on.

And although this consistency and normalcy seem to move around all the time, anomaly detection can do wonders for DevOps and infrastructure-based monitoring, because it really helps with the evolving thresholds of different metrics on your infrastructure.

There is also a great deal of anomaly detection that has been historically used for fraud prevention, in the purest definition of fraud which refers to monetary transactions.

Often when this fraud definition is expanded to include different types of activities in the sense of constitution and on-line system abuse, classification and regression models are developed that better represent the types of behaviors that are trying to be detected. Some good recent examples of companies developing machine learning models and publishing some details on their techniques are AirBnB [11] and Square [12].

There are 3 big issues that have to be dealt with in network anomaly detection for it to be really useful to analysts, and reduce the amount of false positives to a manageable level:

- The “Curse of Dimensionality
- Normality-poisoning attacks
- Hanlon’s Razor

The "Curse of Dimensionality"

This is often referred as one of the biggest issues when we apply anomaly detection to a sufficiently complex problem. It stems from the fact that anomaly detection techniques (or outlier detection) require a measurement (a "distance") to be measured between its different nodes or events so that nodes that are too far can effectively be detected as too distant from an arbitrary level of normality.

The most common distances that are used in this kind of measurement are the Euclidean distance (the "size" of the vector that connect two different nodes in the multidimensional space) or the Manhattan distance (that sum up all the distances in all the different dimensions).

Our intuition on low-dimensional data does not translate correctly what happens into higher dimensional spaces. There are two main results that come from this situation:

- As the number of dimensions increase, the distribution of distances between all pairwise points in the space becomes concentrated around an average distance, so the "sameness" or normality that you are trying to detect start to become meaningless. This is especially true for when you have a great deal of points in the space, where it starts to approximate the uniform distribution in some segments of the space.
- Also, as you try to cluster points within a certain distance of an event, the volume of the high dimensional sphere becomes negligible in relation to the volume of the high dimensional cube. The practical result is that everything just seems too far away, and at similar distances.

Mathematical details from these results can be reviewed here [13].

If you are unsure that we are handling a high dimensional problem here, imagine a company that have n nodes that are able to communicate with each other in the network for a total of 2(n2 - n) combination of communication directions. If we consider TCP and UDP ports separately (as a Netflow based data-source would), we approach something along the lines of 8*65535(n2 - n) measures of network activity we have to analyze for normality. For a very modest company with 1000 nodes, we easily approach half a trillion possible dimensions.

The most promising way to handle with this problem is by having alternative distance measures that project this kinds of data in different spaces or with interesting ways to segment it. A lot of different possible algorithms and techniques to handle that have been tried over the years, including a preceding the clustering attempts with dimensionality reduction (such as PCA or SVD) to assist in finding the biggest outliers in a hyper-dimensional space.

There are a few companies that are based on extensive research results on creating specialized subspaces and distance measures where these kinds of comparisons would work, but this remains a very open research question still. Addressing this challenge would impact various industries, and not only Information Security.

The problem is that not only the biggest outliers are the ones we should be worried about.

Normality-poisoning attacks

One of the biggest truths of Machine Learning of any sort is that your model design, that is, the features you are extracting in order to feed the prediction engine, is of greater importance then the actual algorithms that are being used.

But of even greater importance is what you establish as the ground truth, or what you are using as the baseline of your analysis or model. When you are working with supervised models this is represented by the quality of your labeled data, and how representative it of the elements that you are trying to classify.

And when you are working with unsupervised learning, specifically around anomaly detection, you would need to make sure that whatever normalcy or proper use pattern is not being disturbed. Otherwise, your anomalous behavior is going to be hidden inside what looks like normal traffic.

It soon becomes very clear that if a data source feeds a security decision process, attackers will want to manipulate that data source to its advantage, in a practice that is not unlike clearing your traces in logs after breaking in.

These diversions could either be very discrete and localized, or just outbursts of data to shift the baselines. There would be several ways that these data points could be injected into the system, from additional network traffic, log ingestion to direct manipulation of the datasets of the decision systems. Some interesting ideas and attack frameworks are detailed in [14].

An anecdote I like to share is about Waze (which was recently acquired by Google), a GPS route application where you can report incidents like heavy traffic and accidents, and the crowd sourced input from all their users help tailor the traffic algorithms. Some people I know would report several accidents around their office 15 minutes before leaving, to make sure the traffic from these app users would decrease.

Hanlon's razor

The crux of Hanlon's Razor is *"Never attribute to malice that which is adequately explained by stupidity."*[15]

Given that anomaly detection is essentially an exercise in the unknown, even if you have addressed the other two issues, there is absolutely no guarantee that whatever this super algorithm would be able to uncover is actually a security thread.

If your system uncovers an increase of network traffic between the web servers and the back end database is it more likely that this is happening because you are being attacked by an external assailant or because the new hipster developer that was just hired to your team started experimenting with Node.js on the production servers in order to be "closer to the metal"[16]?

And this of course makes evident one of the usual sins we find in Information Security. Technology for technology's sake, when it is not well wrapped around in process, could actually hinder a security analyst team's ability to respond to actual threats.

This reinforces the earlier point that I made around the usage of anomaly detection tools for DevOps and system management in general. Operations teams would ideally own these processes and do the initial triage if something is to be escalated to the security team.

Potential Solutions

A lot of effort has been put into "profiling" the usage of application and network resources for anomaly detection. This would effectively encode the higher dimensional network information of network traffic into events and actions with meaning for users and assets in organizations.

This efforts have been packaged a lot in marketing materials around "user behavior analytics", where advanced algorithms would effectively detect the the next Edward Snowden before they could steal all your data.

If this sounds suspiciously like DLP, it should.

The main issue for these potential solutions is the structure around the applications, file repositories and the like. How much pre-configuration (classification of information, anyone?) and customization for the internal applications actions to be represented in the system? Wasn't this the model we were trying to move away from?

But there is still potential in this field, especially around the resurgence of the use of association rules in creating the most likely paths of usage in web applications. That is the path that Silver Tail was going through before being acquired by RSA, and is the path that I believe some of these newish ad-bot detecting companies are going through now.

Classification

Classification is a much more defined quantity, and a solution that requires a higher intuition on what exactly is the type of problem you are trying to solve.

Most of the work that I have performed under MLSec Project has been around the use of classification of malicious actors and I have purposefully kept away from straight using anomaly detection, and always looking for ways in what the information could have a high degree of confidence associated to it and providing good conclusions.

Also a lot of the motivation and intuition about the creation of the learning processes and feedback loops in the system are around how incident response has evolved over the last few years and what types of data security analysts have been using in order to make their decisions if something should be investigated or not.

There are two main challenges in classification design:

- Model aging: the threats evolve quickly, as we all know and the patterns that have been encoded into a model a week ago are probably not relevant anymore for detection of some of today's threats. This requires those classification models to be constantly re-trained with fresh labeled data in order to keep up to date. This is mostly an implementation challenge, but it requires some time in batch processing.
- Bad ground truth: similarly to anomaly detection before, a great deal of the academic research has limited itself to using limited samples of public available data, such as Alexa Top 1M and other information of dubious sources.

On this section, I would like to discuss some of the choices me and my team have made as we are building out our use cases for machine learning in security.

Feature Design

A significant driving force in feature selection in my case has been around features that are intrinsic to the actor or entity being analyzed, specially data points that would cost money or resources to an attacker to be able to change or manipulate.

For public IP addresses, there is an profusion of high quality information around the overall structure and topology of the Internet, and a lot of them are available free of charge for research purposes. There are also several pre-packaged libraries for multiple languages that can help you collect GeoIP, ASN and BGP prefix information from those sources.

Similarly, for hostnames and top-level domain information there are Passive DNS and WHOIS repositories that can be used by a fee to provide authoritative information on the history and changes over time on the relationship between those domain names, IP addresses and entities that register them. One of those leading providers of Passive DNS is Farsight Security [17], which is providing the pDNS data we use for our models.

Also, the same considerations can be done for binary artifacts and libraries, considering their structure, function calls, and even shared code that can be inferred by the compilation targets. There is extensive academic research into this specific area and most anti-virus companies have been doing this for many years now without much fanfare (outside if very specific virus related conferences). I am curious to see how those companies compare in large deployments to "turning on the heuristics in AV", which is an interesting source of false positives.

All of these different data points can be mined for features that can compose a detection model. There is not a lot of advice that can be given for proper selection in addition to actually generating them and trying them out on the models that you are trying to build. Depending on the actual behavior you are trying to predict, different features will provide different contributions to the predictive power and it is usually hard to see what they will be beforehand.

Currently on the model implementations at MLSec Project a great number of variations on the features described above are generated and the model-build codes automatically select the ones that have the appropriate variance and distribution to be able to differentiate between the different labels.

Ground Truth

As I said before, the need for appropriate ground truth is actually the most important thing that should go into model design. And when you are looking for labeled data around goodness and badness of specific entities, the volume of information available has increased several times over the last few years.

The selection of the labeled data that will be used to train your model is very important, and should not be limited to just a few public blacklists and the Alexa Top 1M websites. In fact, if this is what you use as your training set, you will only have a model that is effective in separating one group's view of what is malicious from the most popular sites on the Internet. One interesting new source is provided by OpenDNS [18] with their version of the top domain list and a random sampling of non-malicious ones for model training.

Given enough ground truth data from an specific actor or campaign, it should be possible to create models of attribution even, according to the specific patterns that each emerge from different threats, by executing carefully sampled one vs all classification [19].

Conclusion

In closing, the intention of this talk is to be able to clarify some of the advances that have actually happened in this field and try to have organizations and individuals ask the right questions to companies that are developing data analysis and machine learning capabilities as the core of their offering.

Industry and academia have been hot on this topic for almost 20 years, and it might be that it is in generation of companies that we are able to actually deliver some results that will help us in our greatly outmatched battle with attackers.

Make sure you are informed and asking the right questions to your vendors so we can weed out the good and burn down the bad. Feel free to reach out to me at @alexcpsec on Twitter or alexcp@mlsecproject.org if you have an interesting contribution in these research topics.

References

[1] Reducing the Critical Time from Incident Detection to Containment – New ESG Whitepaper (register-walled): http://www.bradfordnetworks.com/resources/whitepapers/reducing-the-critical-time-from-incident-detection-to-containment/

[2] Defending Networks with Incomplete Information: A Machine Learning Approach: https://media.blackhat.com/us-13/US-13-Pinto-Defending-Networks-with-Incomplete-Information-A-Machine-Learning-Approach-Slides.pdf

[3] Next-Generation Intrusion-Detection Expert System (NIDES) - http://www.csl.sri.com/projects/nides/

[4] MIT Lincoln Laboratory - DARPA Intrusion Detection Data Sets http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/index.html

[5] Solaris Basic Security Mode (BSM) Auditing - Hal Pomeranz - http://www.deer-run.com/~hal/sysadmin/SolarisBSMAuditing.html

[6] Google Scholar Search for "MIT DARPA Intrusion Detection Evaluation" - http://scholar.google.com.br/scholar?q=MIT+DARPA+Intrusion+Detection+Evaluation

[7] KDD Cup 1999 Data - http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[8] Google Scholar Search for "KDD Cup 99" - http://scholar.google.com.br/scholar?q=KDD+Cup+99

[9] John McHugh - Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory - http://people.scs.carleton.ca/~soma/id-2006w/readings/mchugh-darpa.pdf

[10] Anton Chuvakin - SIEM Analytics Histories and Lessons - http://blogs.gartner.com/anton-chuvakin/2014/06/06/siem-analytics-histories-and-lessons/

[11] Naseem Hakim and Aaron Keys - Architecting a Machine Learning System for Risk - http://nerds.airbnb.com/architecting-machine-learning-system-risk

[12] Rong Yan - Square's Machine Learning Infrastructure and Applications - http://www.hakkalabs.co/articles/squares-machine-learning-infrastructure-applications

[13] Excerpt from "CS 4850 Mathematical Foundations for the Information Age" materials from Cornell University -

http://www.cs.cornell.edu/courses/cs4850/2010sp/Course%20Notes%5C2%20High%20Dimensional%20Data-Jan-2010.pdf

[14] Barreno, M. et al - The security of machine learning - http://bnrg.cs.berkeley.edu/~adj/publications/paper-files/SecML-MLJ2010.pdf

[15] Wikipedia - Hanlon's razor - https://en.wikipedia.org/wiki/Hanlon's_razor

[16] Hacker News - Comment by NHQ - https://news.ycombinator.com/item?id=3965392

[17] Farsight Security - https://www.farsightsecurity.com/

[18] Github - OpenDNS Top Domains List - https://github.com/opendns/public-domain-lists

[19] Wikipedia - Multiclass Classification - https://en.wikipedia.org/wiki/Multiclass_classification

MLSec

MACHINE LEARNING SECURITY