LIGHTCYBER

CLOSING THE BREACH DETECTION GAP

MACHINE LEARNING

Unlocking the True Potential of Machine Learning to Enhance Network Security

WHITE PAPER / MAY 2016

Executive Summary

Is Machine Learning (ML) poised to transform the security industry? To hear the claims of vendors, the answer is a resounding yes. Unfortunately, the reality is much more nuanced. While machine learning can improve techniques for detection, ML alone will not transform security if its only use is to incrementally improve security tools in their current pursuit: the one-dimensional focus on the tools of attack. What is needed instead is to augment current and next-generation prevention technologies (like firewalls and endpoint protection platforms) with robust detection. This combination of conventional prevention with innovative ML-based detection and response is the new form of "defense in depth," and it is in this pursuit that ML unlocks fundamentally new capabilities that were not previously feasible.

This paper will elaborate on the differences between known bad and known good security models, then explore a common-sense framework for understanding how Machine Learning can be employed to transform information security. Finally, the paper will empower the reader to quickly differentiate between vendors throwing around buzzwords in pursuit of a futile objective (perfecting known bad), vs. those on a worthwhile path of complementing existing prevention infrastructure with a new approach to detection.

> "Machine learning or data science, despite the popularity of these buzzwords in the marketing materials of security vendors today, is only as good as its application. Applied to the wrong problems, or in the same ways security has been approached for 20 years will deliver the same results: excessive false positives, and the continued inability to detect critical attack activity."

Known Bad Has Reached Its Useful Limit

The vast majority of security tools brought to market over the last 2 decades have been focused on a "known bad" model. This means understanding and describing the tools or techniques a malicious actor uses, e.g., Hashes for viruses, reputation for domains, IoC's for more advanced malware analysis, etc. This model has proven its limits; attackers continue to run rampant in our networks, often undetected for months if not years, while security organizations are overwhelmed with alerts and alarms, yet unable to effectively detect attackers within their networks: a recent Ponemon survey reports the average organization receives 16,937 alerts per week, and as a result only able to investigate 4% of alerts.

Machine learning is NOT poised to fundamentally change this. Machine learning or data science, despite the popularity of these buzzwords in the marketing materials of security vendors today, is only as good as its application. Applied to the wrong problems, or in the same ways security has been approached for 20 years will deliver the same results: excessive false positives, and the continued inability to detect critical attack activity.

What is needed is a fundamentally new approach; a "known good," or "learned good" model. Prevention based on known bad will continue to be necessary, but is no longer sufficient. It must be complemented by accurate and efficient detection, but a detection approach that doesn't rely on the same techniques of known bad. Rather than attempting to enumerate the techniques used by attackers, a known good model should seek to learn or baseline the normal activity or users or devices and detect the anomalies that are attack-relevant.

Unfortunately, this is no mean feat. Networks are incredibly diverse and complicated, and no approach to date has been able to tackle this problem effectively. This is where machine learning can have a dramatically different result, but the tools of data science must be employed in the right ways.

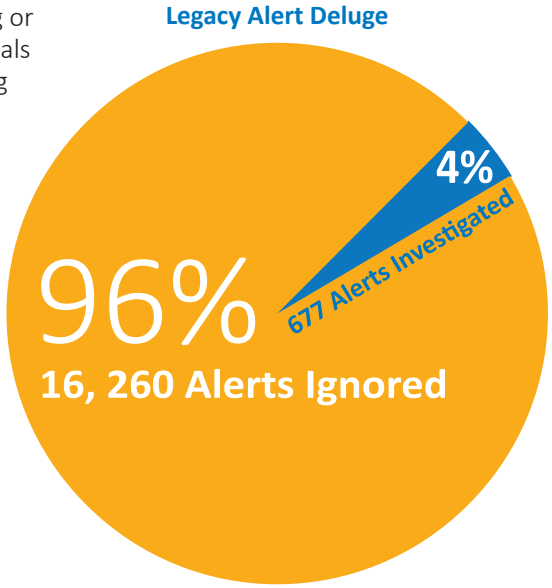


Figure 1: More alerts than anyone can handle, and most un-examined.

Machine Learning: A Common Sense Approach

While a valid approach to this discussion would be to dissect the techniques of machine learning (regression, classification, neural networks, feature extraction, deep learning, etc.), such an approach is best suited for academia and those with an in-depth knowledge of ML techniques. Instead, a simple observation can be made and acted upon: Machine Learning is only good for the tasks it is trained on, based on the inputs and feedback it receives. Therefore, understanding the inputs to ML systems can go a long way towards understanding what problems they are good for.

What we propose is a journalistic approach to understanding how ML is being utilized by products:

- **WHO** performs the analysis
- **WHAT** is analyzed
- **WHERE** is the analysis performed
- **WHEN** is the analysis performed
- **HOW** is ML deployed (this is the more technical bit, skip if desired)

This can then inform on the outcome: what kinds of results is the ML system designed to produce? Are such ML results a good complement to existing technologies, or just more of the same? After all, all the traditional anti-virus vendors have been utilizing statistical and machine learning based engines for years — while it is possible a startup with fancy marketing has figured out how to move the needle a bit on malware detection, such approaches will still never by 100%, and will remain 0% effective in finding non-malware based attacks.

Who

Who is performing machine learning. Is it the vendor's research team and data scientists, or is the product itself performing the machine learning. This simple question alone can strongly differentiate product approaches.

What

What is being analyzed? What are the inputs? If the inputs are simply labeled samples of malware or other objects (files, registry changes, IoC's, domains, etc.), then whether what is being analyzed are the files themselves, or the activity they generate to the Internet (or to DNS resolvers), the only result will be an engine for detecting malware. On the other hand, if what is being analyzed is part of the production systems of your organizations (network traffic, endpoints, or logs), then the system is likely designed with something other than simple malware detection in mind.

Where

Where is the analysis performed? Analysis in the vendors' lab indicates that the system isn't going to be learning much about your network or users at all. Analysis performed on your site strongly implies the system will be learning and profiling some aspect of your environment. This is the key to learning or knowing what is normal on the network. With a small bit of feedback to the system, this can become a known good model and power anomaly based detection that highlights attack behavior, rather than simply looking for more objects.

When

When is the analysis performed? If learning takes place before the product is deployed, then clearly it is not learning anything about the live network where it is deployed. Learning in advance, by definition, must be learning only about known bad or related objects. Learning that happens after deployment is likely focused on the activities that are happening in real time on your network as your users go about their normal tasks.

How

Supervised machine learning techniques are defined as those that take as inputs labeled (known) samples of (for example) malware vs. benign files. These techniques are best used to learn and thus describe and identify what they are trained on. Unsupervised ML techniques are defined as where there is no predefined set of known examples, and instead the system must group ("cluster") and infer. The devil is really in the details here, however, because all decent ML implementations actually blend many (if not hundreds) of different models, and can actually extract learned features from one problem space and sometimes even effectively apply them to an adjacent or similar problem. This is why the "how" is not the best mechanism for understanding whether a vendor's approach will be beneficial or not, but a basic understanding of at least whether the approach leans towards supervised or unsupervised can help clarify whether the answers to the rest of the questions indicate a solution focused on "known bad" or is tackling detection from a new angle.

Why does any of this matter? Because you can't hope to solve your security problems with a tool trained on the wrong task! The next step might be to ask - what are the benefits of focusing on known bad vs. known good, and then how do I tell whether a vendor is going to help me with one or the other?

Machine Learning Framework: Location Matters

Who	**Vendor**	**Product**
What	**Objects (files, etc.)**	**Activity (network flows, etc.)**
Where	**Vendor Lab**	**Poduction Systems**
When	**Before**	**During**
Why	**Signiture 2.0**	**Learned Baseline**
How	**Supervised machine learning (labeled samples of known good and bad)**	**Combination of unsupervised and supervised ML techniques**
So What:	**Known Bad (2.0)**	**Known Good**

Figure 2: The Who-What-Where… of Machine Learning: Known Bad vs. Known Good

Known Bad

Known Bad focuses on identifying the technical artifacts (i.e., "objects") attackers use such as malicious files (malware), domains, registry changes, IP's, etc. The major benefit of the Known Bad approach is that with many techniques, the identification of a Known Bad object can be performed quickly enough that the attack (step) can be prevented, i.e., a virus can be stopped from executing, a connection to a bad domain can be blocked, etc. This is great when it works, and we of course want to prevent as much as possible; but unfortunately, we do not live in an ideal world where prevention can work 100% of the time.

The two-fold tradeoff of Known Bad:

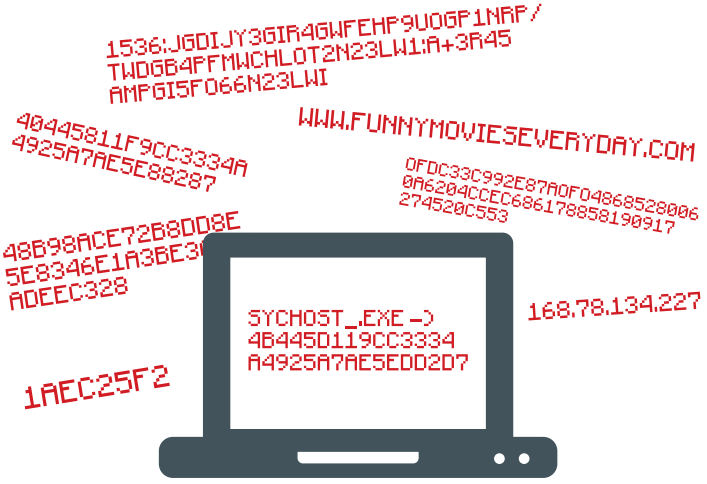
First, such solutions will never be able to deal with the increasing amount of attack activity that utilizes the concept of "living off the land." Such approaches mean that attackers don't utilize malware or malicious tools, but instead use the standard set of (usually very powerful) IT tools and the services they can access with stolen or compromised credentials. Similarly, attacks initiated with legitimate credentials by malicious internal users would never be identified by such solutions.

> "The major benefit of this approach is that it is not stuck in an arms race with attackers, and what's more, attackers cannot possibly test their techniques against a known good model."

Second, even powered by ML, there is still a never-ending arms race that security vendors are generally on the losing end of. Attacker can continue to innovate their tools, and what's worse, they can test them against known-bad prevention or detection products in advance to ***guarantee*** the efficacy of the tool they want to use in advance of the attack!

These two facts point to the inevitable need for something else.

Known Bad: Objects



Known Good: Behaviors

Figure 3: Known Bad concentrates on objects like file signatures, domains, etc., whereas Known Good focuses on behaviors to alert on anomalous attacker activity.

Mapping Vendor Claims to Known Bad / Known Good

Known Bad:
- Inputs are files (malware) or other objects
- Analysis is done in the vendor lab.
- Product claims real-time detection capability
- Product does not require a learning or soak period, or the majority of the detections of the product are immediately available

Known Good:
- Inputs are real production data: network packets, endpoint information, and/or logs
- Analysis is done on-premises
- Product does not claim real-time detection, but instead focuses on behavior and anomaly
- Product requires a learning or soak period before the majority of detectors are online

Figure 4: How to differentiate ML approaches based on vendor marketing claims

Known Good

Known Good focuses in this context not on simply whitelisting of authorized files or activities, but more generally the ability to learn or baseline the normal activity of users and devices. Alone, this cannot power detection. But combined with an ability to continuously monitor for attack-relevant anomalies, this can provide insight into an entire range of attack activity to which organizations are otherwise blind.

The major benefit of this approach is that it is not stuck in an arms race with attackers, and what's more, attackers cannot possibly test their techniques against a Known Good model. This is because they cannot know the normal behavior or use of systems before their arrival in a target environment, and the very process of learning is such that they will create anomalies that can be detected.

The tradeoff is that such behavioral anomaly systems cannot usually trigger on any single event, but instead must observe a pattern of activity over time. This generally precludes an inline blocking mode, and instead is best used to drive a model of detection and response.

Such Known Good models were not feasible prior to the development of recent Machine Learning techniques that automate the learning and detection processes.

Not all behavioral detection systems are the same, even if similar ML techniques are utilized. A large class of solutions focus primarily on having logs as the input. While this can be appealing because of the cost already sunk in gathering and retaining logs, there are a number of tradeoffs. The most serious is that such solutions look for attack and insider threat behaviors based on log data about user actions, because credential (login) information is often the best/only security relevant data available in such systems. Unfortunately, this approach misses the network and device behavior which can trigger alerts earlier in the attack lifecycle. If you have to wait until credentials are compromised and used to spread, it means you missed command and control, network scans, changes in device behavior, etc. before the attack got to that stage. Network-based approaches can capture all of this information, and thus ML systems that are learning network activity have a broader ability to detect attack activity than any other option.

Machine Learning – Highly Input Dependent

Who	Product	Product
What	Activity (network flows, etc.)	Record of Activity (Logs)
Where	Poduction Systems	SIEM
When	During	During / Late => After
Why	Learned Baseline	Learned Baseline
How	Combination of unsupervised and supervised ML techniques	Unsupervised - threshold-based rules
So What:	Known Good	Known Good

Figure 5: Examining the Who-What-Where-When-Why-How of ML

As common sense would indicate, more is often better. This is true here too. Network may give great depth, but endpoint can power a lot of specificity that really drives accuracy and actionability. A solution that starts with the network, profiling users and devices based on metadata extracted from network traffic, will have the best broad detection capabilities. Adding the endpoint adds the ability to detect low prevalence files and processes, and if done right can be used to associate the process with the network activity (something difficult to impossible to do from the network alone) to actually associate a detected network behavior (such as a port scan or command and control traffic) back to the originating process on the workstation, and the user account that started the process. This functionality provides added context and validation which speeds up the time of detection and reduces the amount of time that analysts need to understand and make a determination about a given situation.

What we need to remember is that ultimately these detections will be delivered to a human user for action, so the the goal should be to provide the most accurate detections (with as little noise as possible) augmented with as much information as possible to enable swift triage and response.

> “Instead, it is probably better to pursue a strategy of complementing existing security infrastructure investments with a new class of detection. The distinction is not whether ML is used or not, but instead is it used to power a new approach to the problem: a system for learning the known good behavior of users and devices, and detecting the anomalies that attackers introduce (regardless of tools and techniques).”

An Aside On Inputs

Inputs really matter for both ML, and detection as a whole. While so far we've focused on the differences between analyzing objects like files (malware) vs. real production data, it is also worth examining the options for building a known good model. The choices boil down to:

- Network traffic (packets)
- Endpoint information
- Logs (generally as aggregated in a SIEM)
- Or some combination of the three

Network

- Attacker cannot evade or conceal
- An attack that doesn't use the network isn't an active threat
- Amazing depth and breadth of information
- Can be challenging to architect access

Endpoint

- Where attacks originate
- Points of infection if malware is used
- Challenging to operationalize, especially across non-pc devices
- Often pigeon-holes on malware, vs. higher level user behaviors
- First thing attacker does is disable agents

Logs

- Already centralized, large investment in SIEM
- Normalized, subject to source log level settings and ingest process
- Reduced breadth of visibility, generally constrained to credential access only, not full attack inputs
- Not real-time activity, after-the fact generated record
- Second thing attacker does is disable logging.

Conclusion

So, if you want to buy a product from a startup claiming ML is powering a new form of what is ultimately known-bad detection or prevention, know that these are the same techniques that all the major players have been working on for years. All the NGFW and major EPP players have been adding these capabilities to better detect malicious objects. While there may be an incremental gain in blocking available, the fundamental truth remains: prevention and even known-bad detection will never be 100%. Instead, it is probably better to pursue a strategy of complementing existing security infrastructure investments with a new class of detection. The distinction is not whether ML is used or not, but instead is it used to power a new approach to the problem: a system for learning the known good behavior of users and devices, and detecting the anomalies that attackers introduce (irrespective of tools and techniques).

Find out How LightCyber Can Work for You to Lower Alerts and Increase Security

To learn how LightCyber Magna can help you improve security by accurately and efficiently locating and remediating attackers on your network, please contact us to schedule a customized demo.

About LightCyber

LightCyber is a leading provider of Behavioral Attack Detection solutions that provide accurate and efficient security visibility into attacks that have slipped through the cracks of traditional security controls. The LightCyber Magna™ platform is the first security product to integrate user, network and endpoint context to provide security visibility into a range of attack activity. Founded in 2012 and led by world-class cyber security experts, the company's products have been successfully deployed by top-tier customers around the world in industries including the financial, healthcare, legal, telecom, government, media and technology sectors. For more information, please visit http://www.lightcyber.com.

LIGHTCYBER
5050 El Camino, Suite 226
Los Altos, CA
Ph: (844) 560-7976
www.lightcyber.com