WHITE PAPER

Benefits of Machine Learning

with Behavioral Analysis in Detection of Advanced Persistent Threats

CYPHORT™

Overview

The Evolution of Advanced Persistent Threat Detection

Computer viruses have plagued personal computers since the original Brain virus began infecting boot sectors in 1986. Originally, these early viruses were annoying, but fundamentally benign in nature. However, once the initial concept of malicious propagating code became established, the actors creating viruses became more sophisticated in their approach. Ultimately, the results of a successful infection were more significant and the impact on an enterprise more severe.

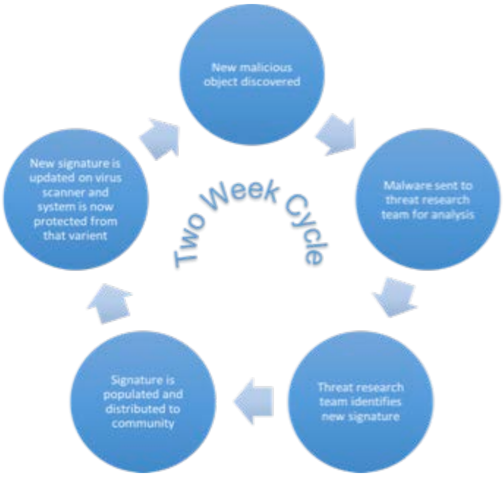
The initial solution to these threats came in the form of virus scanners. Organizations that created early virus scanners would identify new viruses, analyze the malicious code, and isolate the commonly recognizable patterns in the virus that could be used as a “signature” to identify the virus on protected systems. Virus scanners would then use these signatures to analyze all files on a system and alert users when a known infection was introduced onto the system.

While this solution proved to be effective for early viruses, it was not without its complications. First, when a new instance of viruses was discovered in the wild, it needed to be identified, sent to a security research team, and analyzed to develop the new signature. This new signature needed to be distributed to each system. This cycle could take several weeks, during which time the virus would be able to act and spread unchecked throughout organizations. The second challenge was the ease in which minor changes could be made to the malware. These changes did not fundamentally change its behavior, but rendered the original signature ineffective against these new variants.

As malicious actors grew in sophistication, basic early viruses gave way to more sophisticated malware. Initial efforts to manually change their malware to evade detection ultimately yielded techniques that allowed that malware to automatically “mutate” following each successful infection. These mutations allowed malware to propagate unchecked for even longer durations while detection techniques were developed.

By the time Stuxnet was discovered in 2010, it became clear that a new breed of malware had arrived: the Advanced Persistent Threat, or APT. APTs are defined as malware that:

- Employs obfuscation techniques to evade detection.
- Attempts to create a persistent “beachhead” in order to maintain access to an organization following an attempt to eradicate it.
- Is frequently used as a targeted attack and has a specific goal, whether that is financial gain, data exfiltration and espionage, or simple sabotage.
- Evades virus scanners.


New malicious object discovered
Malware sent to threat research team for analysis
Threat research team identifies new signature
Signature is populated and distributed to community
New signature is updated on virus scanner and system is now protected from that varient
Two Week Cycle

First Generation APT Detection

APTs are difficult or impossible to identify using signature-based analysis techniques alone. The nature of APTs is to rapidly modify its characteristics so that the many variants will not be identifiable. Even if one instance of the APT is identified, a single organization may be targeted with several variants to ensure the successful deployment of at least one instance. As identification through signature was no longer reliable, a new technique needed to be created. Thus, the concept of sandboxing was introduced.

Sandboxing is a relatively simple concept. Rather than identify malicious code by signature, the detection system creates a virtualized environment that appears to the malware to be an ordinary PC workstation. The tested object is opened in this protected environment and its behavior observed. If a document, for example, was opened, and was then seen to download additional payloads, make changes to the Windows system, or display other abnormalities, it is clear that the file is not operating as expected. Observing the behavior of a document in a safe, partitioned environment allows the user to identify APTs by their behavior, not static characteristics that would be simple to obfuscate.

Identification of malware by behavior analysis proved to be a very effective strategy. Unfortunately, this development has just advanced the cat-and-mouse exchange between enterprises and the adversaries creating the malware. As sandbox analysis toolkits became more prevalent, APTs were created to identify when they were operating in a virtualized environment. Once malware is created to search for characteristics unique to one sandbox, the sandbox vendors alter the sandbox to obfuscate these traits in order to encourage malware to demonstrate the full spectrum of behavior. Most importantly, APT authors have discovered that by altering their techniques, sandboxes designed around rules-based techniques for identification can be evaded.

Challenges with Rules-based Analytics for Sandbox Analysis:

- Manual process of creating malware rules requires heavy investments in malware research teams. Even with a large team of threat analysts, organizations are finding it hard to keep up with the increased variety and volume of threats.
- Threat analytics rules start to lose effectiveness as soon as as they are released since malware authors can redesign their malware to skirt around these rules with minimum effort.
- Malware analysis today requires understanding of thousands of subtle malware behaviors making manually coded rules impractical and ineffective in finding advanced attacks.

If a rules-based sandbox is programed to look for a pattern where X happens, then Y and then Z, the APT developers can change the behavior to evade such rules-based techniques. When such changes occur, the malware must be sent to the security research team, who will then learn how to tweak their rules to identify this new variant. These new rules can then be pushed down to the sandbox and future APTs using this new technique can be identified. This method is dangerously close to the failed processes that made virus scanning ineffective. Once again, the industry has created a detection system where new variants of APTs using new techniques can require a two-to-three week turnaround time between the initial discovery and the ability to effectively protect against the new technique. While the effort in changing the APT to avoid detection is much higher and requires greater sophistication, the rewards for a successful targeted attack make the labor seem worthwhile.

Next Generation Analysis

The complication with first generation APT detection is in the degree of manual labor that is required to deliver the agility necessary to protect enterprises from the current generation of threats. First generation APT products have replaced the exercise of analyzing new variants of a threat to generate a virus signature, with an exercise of analyzing new variants of a threat to generate new rules for their sandbox detection engine. With either resulting product, a two-to-three-week gap between discovery and protection which is enough to cause irreparable harm to an enterprise.

The problem with the existing solution is that user input doesn’t scale. Any solution that requires a skilled analyst to be integral to what is meant to be an agile process is destined to be the source of delay. This is exactly what the new discipline of machine learning was designed to solve. Machine learning is a style of artificial intelligence that enables systems to learn relationships through the analysis of a large dataset.

Next Generation APT analysis relies upon machine learning instead of a more rigid rules-based system of analysis. Machine learning works through the analysis of large datasets in order to identify correlations through the observation of patterns. The larger the dataset, the greater is the possibility of statistically relevant results. To use this technique in the identification of malware and APTs, these file objects are permitted to execute in an isolated sandbox environment that is instrumented to monitor the behavior for the duration of the test.

Benefits of Machine Learning Analytics for Advanced Threats Detection:

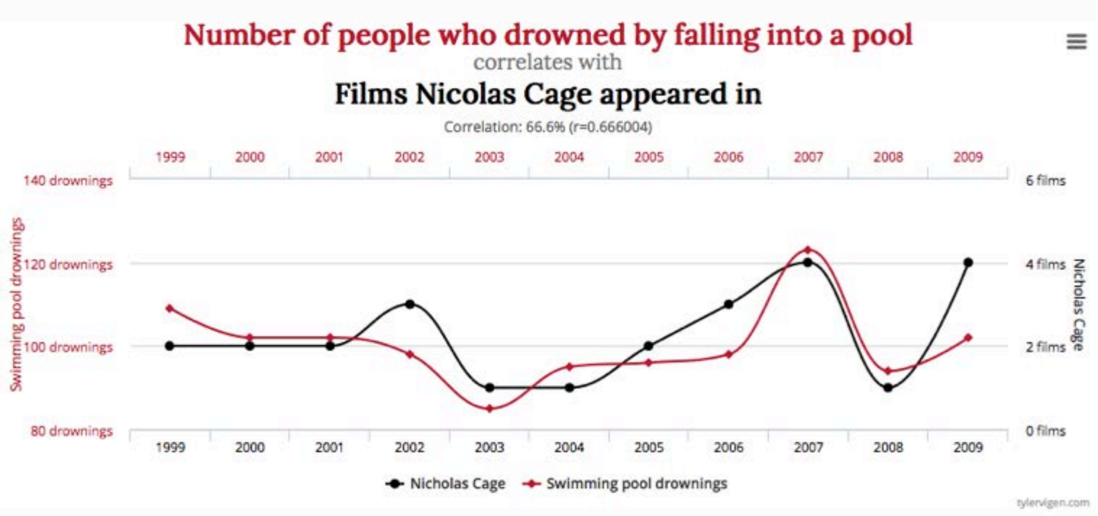
- Instead of armies of malware analysts manually analyzing and encoding malware detection rules, a small team of highly skilled data scientists can keep the machine learning model updated to deal with the latest threats.
- Detection mechanism continuously evolves as it finds more threats and learns about the nuances of malware behavior.
- Minor behavior patterns are recognized by the analytics engine, compared to a rules-based system where a human analyst can only encode a behavior rule they can observe.

The behavioral logs themselves constitute the basic building blocks for analysis. To provide the data necessary for statistically relevant results, behavior logs from hundreds of millions of file objects, some malicious and some benign, are compiled into a database that can be analyzed. Using this data and understanding which objects are malicious and which are benign allows for sophisticated models to be created that will enable the system to categorize objects simply by correlating analysis results with the ever-evolving models.

The results of this strategy have proven to deliver a high degree of precision. They have also been useful in automating the process of not only identifying which objects are benign or malicious, but also in identifying malicious objects with even greater granularity by categorizing the type of malware the object most closely resembles. The system can identify adware, ransomware, and other target-intent malware with a much higher degree of accuracy than first generation APT detection.

Effective Use of Machine Learning

Machine learning is the new hot topic for any solution that hopes to harvest “big data” for the purpose of solving difficult problems that require detailed analysis. Unfortunately, while the tools for achieving this have become more publically available, the knowledge of how to use these tools effectively is still beyond the grasp of many. To illustrate this problem, Tyler Vigen published a book titled “Spurious Correlations[1],” where he demonstrated the correlations between the number of films in which Nicolas Cage appeared annually with the number of people who drowned by falling into a pool. Correlation clearly does not imply causation.



[1] Vigen, Tyler. Spurious Correlations. Hachette, 2015. Print.

While this demonstrates how not to use machine learning, it is all too common for people unfamiliar with statistics to use this new technology to correlate datasets with little or no association. In such cases, the results will be questionable at best. Machine learning can rapidly produce incredible results, but it can also be used to support results that have little foundation in reality.

There are two basic categories of machine learning: supervised and unsupervised. With unsupervised machine learning, the system is thrown a large volume of data and is asked to discover relationships. While this technique is interesting from a discovery standpoint, its primary value comes when the user is not approaching the system to solve a specific problem but rather to learn more about the datasets themselves. Supervised machine learning, by contrast, is used to solve a more specific problem.

Supervised machine learning begins with a “predictor function”, and sets of data points that may be used by the predictor function. This predictor function can be optimized by training a model against the large labeled data sets, and ultimately measured against a “known value”.

For the purposes of APT detection, a system would analyze files and store a result log for each examined file that describes each behavior the object exhibited during analysis. Creating a machine learning model involves analyzing the results of hundreds of thousands of malicious files and hundreds of millions of benign files. The “known value” being measured against whether an object is malicious or benign. A model can then be created where the behavior logs most closely result in a precise verdict.

It is important to understand which features of the dataset can be used as relevant input values to the predictor in order to produce the desired results. Many of the behaviors the objects exhibit may be completely irrelevant to the purpose of identifying malicious intent. Others may only serve as indicators when they occur at a given frequency. Differentiating the signal from the noise in this respect is one key to creating a precise model.

Any system that identifies a file as malicious or benign through the observation of behavior has a degree of precision with which it is associated. In this field, there is no such thing as accuracy, just precision. With a properly trained machine learning model, the results of file analysis can be compared with the model to determine if the results are within the boundaries of what is known about malicious objects.

Determining where those boundaries should be placed is another key to creating an effective model. The goal when producing a model is to minimize, as much as possible, false positives (identifying something as malicious when it is actually benign) and false negatives (identifying something as benign when it is actually malicious). The more sensitive you make the model, the more prone you are to false positives. You can train a model to have no false positives, at the risk of creating the opportunity for more false negatives, and vice versa. Striking this balance requires large datasets and consistent training as the landscape of new malware and malicious threats changes constantly.

This process can be visualized using a method called the “confusion matrix”. In the table below, there are a total of five malicious objects, and 10,001 benign objects. The model

that created this matrix correctly identified 4/5 of the malicious objects and 1/10001 of the benign. When training the model, there will always be tradeoffs between sensitivity and false positives. It may be possible to make the model that created the table below identify 100% of the malicious objects, at the risk of increasing the number of false positives in the process.

		Predicted	
		Malicious	Benign
Actual	Malicious	4	1
	Benign	1	10000

This can often be influenced by the prevalence of a particular type of threat “in the wild”. In a hypothetical example, consider a model that can be created to detect malware that affects icon files (.ico), and this model was able to detect malicious intent with a 0.0001% chance of false positive. These files are relatively common, but there is no known malware in existence for this kind of file. Therefore, one in every 1,000,000 times this kind of file was observed, a false positive would be generated; and no actual positives would ever be seen because there are no actual examples “in the wild.” In machine learning terms, this is identified as the “false discovery rate”. Ultimately, these results would be ignored as noise. While this is an extreme example, the prevalence and frequency of a threat must be accounted for when training the model so that the number of false positives do not overshadow the number of actual detections. The tolerances for a model that identifies infrequent malware variants must be significantly tighter than those of more common specimens.

Ultimately, it is vital to take many considerations into account when using machine learning in a security context.

1. You must consider the source and the value of the dataset used to train the model.
2. You must strike the appropriate detection balance.
3. You must be able to accurately differentiate the signal from the noise.

Conclusion

The evolution of initial computer viruses to modern APTs has required a significant change in the approach in how these threats are identified. As threats have become more sophisticated, they have developed the ability to obfuscate their behavior to evade previous generations of detection tools. While the original practice of behavior analysis hoped to bridge the gap of time between discovery and protection, modern APTs have discovered methods to reestablish this gap.

Machine learning can be an effective tool in the identification of this new breed of APTs, by establishing more agile practices that allow the system to respond more rapidly to changes in approach and variations in methodologies. However, this technology must be implemented effectively, with the understanding of the relationships of the data features, and understanding of the malware landscape. When irrelevant data is fed to the predictor function, results of the model will be affected and precision will be diminished.

Ultimately, user input does not scale. When a system is created that requires skilled human intervention, the human ultimately becomes the weakest link in scale and performance. However, implementing a machine learning system to solve this problem without clear understanding of the problem being solved will result in less than precise results, which in turn diminishes the effectiveness of the solution. Effective use of an APT solution that is hardened by machine learning implemented in an optimized manner can provide agile, responsive and precise detection of current and future generations of threats to an enterprise being targeted by more modern approaches and sophisticated actors.

Cyphort is an innovative provider of Advanced Threat Protection solutions that deliver a complete defense against current and emerging Advanced Persistent Threats, targeted attacks and zero day vulnerabilities. Cyphort's unique software solution delivers complete Next Generation APT defense; correlating threats from Web and Email, as well as internal vectors for lateral spread and post breach activity detection. The addition of correlation and context for each threat plays a key role in helping prioritize responses and containment actions. The purpose-built integrations to existing enforcement solutions (Firewalls, Proxies etc.) allows customers the ability to extract maximum value from IT assets without compromising the security of an organization. Cyphort is a privately held company headquartered in Santa Clara, California. For more information, please visit www.cyphort.com and follow us @Cyphort.

CYPHORT, Inc.
5451 Great America Pkwy
Suite 225
Santa Clara, CA 95054
P: (408) 841-4665
F: (408) 540-1299
www.cyphort.com