VECTRA
SECURITY THAT THINKS

WHITE PAPER

# The AI Behind Vectra AI

## TABLE OF CONTENTS

**Vectra® protects business by detecting and stopping cyberattacks.**

Vectra® is a leader in threat detection and response for hybrid and multi-cloud enterprises. The Vectra platform uses AI to detect threats at speed across public cloud, identity, SaaS applications, and data centers. Only Vectra optimizes AI to detect attacker methods—the TTPs at the heart of all attacks—rather than simplistically alerting on "different". The resulting high-fidelity threat signal and clear context enables security teams to respond to threats sooner and to stop attacks in progress faster. Organizations worldwide rely on Vectra for resilience in the face of dangerous cyber threats and to prevent ransomware, supply chain compromise, identity takeovers, and other cyberattacks from impacting their businesses. For more information, visit vectra.ai.

## Introduction

Data science is Vectra AI's north star. We have always believed that data science and AI, if used properly, can transform our fight against cyberattacks and give an edge to defenders. However, not all AI is the same. In this paper, we will survey what AI is and explain the key terms relevant to AI solutions. We'll also characterize the two dominant methodologies for applying AI to threat detection and present a deep dive into how Vectra surfaces threats with AI.

Whether you're an AI skeptic or deeply fascinated by the potential of AI, this paper is for you.

## What is AI?

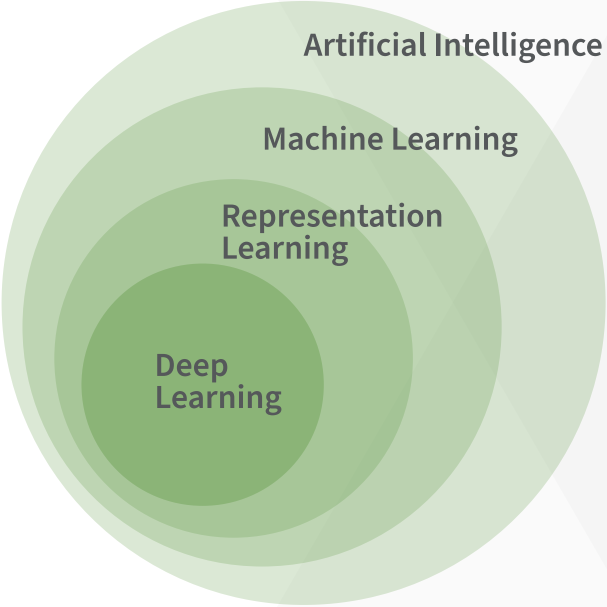
### Defining AI

The terms Artificial Intelligence, Machine Learning, and Deep Learning are often misunderstood as referring to the same discipline or as existing on some spectrum of quality, this is however not the case. These terms are related but each term has its own distinct and specific meaning. Understanding the scope of these terms can allow for a better understanding of what tools are doing that leverage AI.

**Artificial Intelligence (AI):** Artificial intelligence is defined as any system which can automate reasoning and approximate the human mind.  It is a broad and encompassing term that includes the sub-disciplines of Machine Learning, Representation Learning, and Deep Learning. The term AI applies equally to a system that relies on the use of explicitly programmed rules and one that has autonomously gained understanding from a sea of data. The later form of AI, one that learns from data, is what underpins technologies like self-driving cars and virtual assistants and falls into the sub-discipline of Machine Learning.

**Machine Learning (ML):** Machine learning is a sub-discipline of AI wherein the actions of the system are not explicitly dictated by a human but instead learned from data. These systems are capable of processing dozens to billions of data points, to learn how to optimally represent and subsequently respond to new instances of data.

**Representation Learning (RL):** Representation learning, while not commonly discussed, is core to many AI technologies used today. This sub-discipline is focused on learning a new abstract representation from data. An example of RL would be the transformation of images of varying sizes into a list of numbers with a consistent length which represents a distillation of the original images. This abstraction primarily enables downstream systems to better act on new types of data.



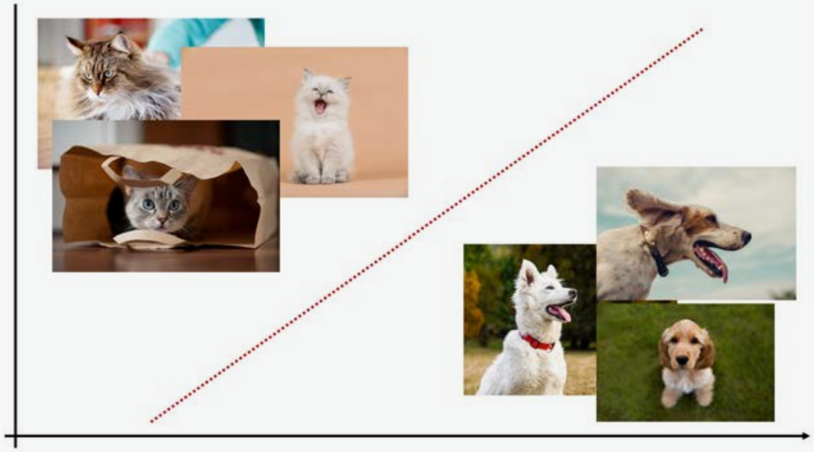
The relationship between different AI sub-disciplines.
Reference: “Deep Learning,” Goodfellow, Bengio & Courville (2016)

**Deep Learning (DL):** Deep learning, often associated with neural networks, builds upon the broader sub-disciplines of ML and RL by discovering from data a hierarchy of abstractions that represent inputs in an increasingly complex manner. Taking inspiration from the human brain, DL models make use of layers of neurons whose synaptic weights adapt in response to inputs, with deeper layers in the network learning new abstract representations that simplify tasks like categorizing an image or translating a piece of text. While deep learning can be an effective technique for solving certain complex problems, it is by no means a panacea for automating intelligence.

## The types of algorithm learning techniques

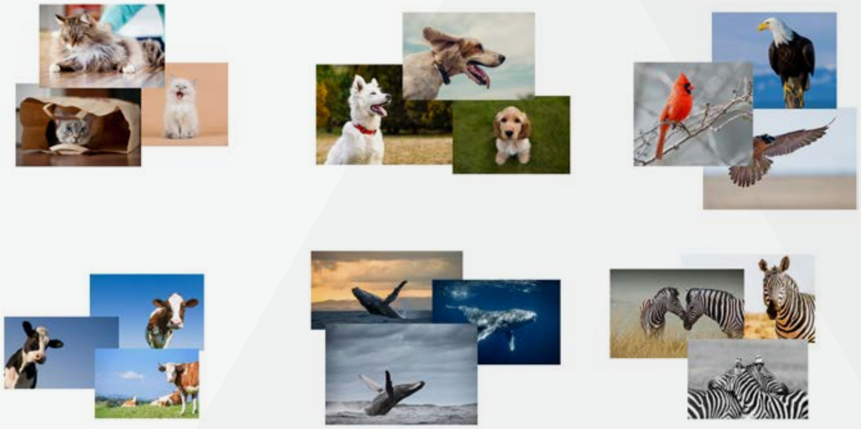
One of the core capabilities of ML algorithms is the ability to classify instances of data into different classes. There are a handful of broad categories of learning that support this capability—the two most prevalent being **supervised** and **unsupervised**.

**Supervised** learning is where the model learns from a set of labeled data. Once learned, given any new data, the model can predict a label. Take the example below – if we feed a supervised learning model numerous images of cats and dogs, given a new image it will predict whether it is a cat or a dog. Supervised learning requires a large corpus of labeled training data for the model to learn from, but once trained these models can be highly effective at generalizing and successfully labeling new instances of data.



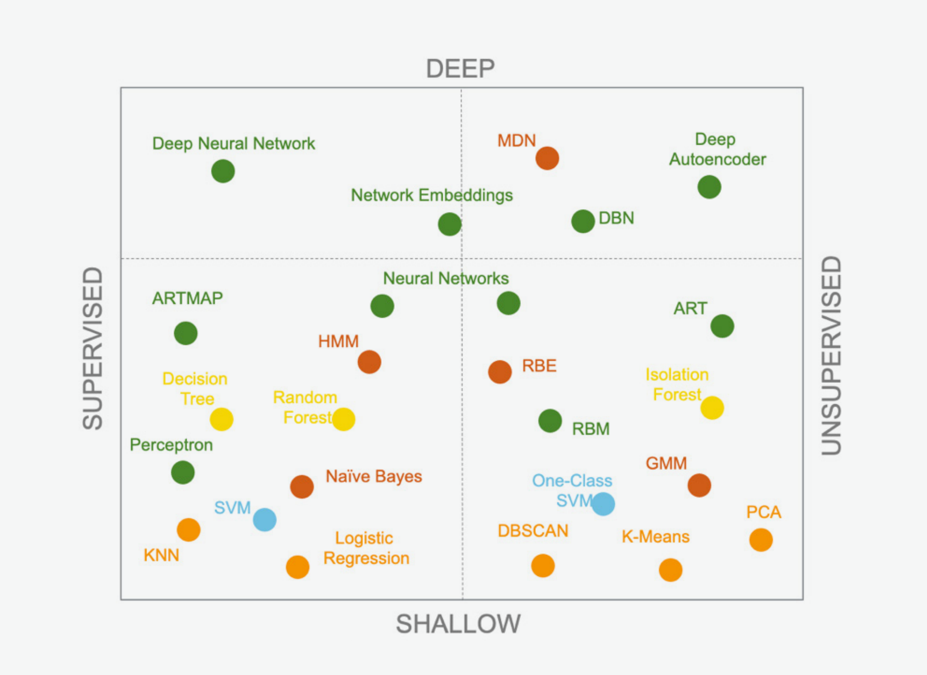
Supervised learning uses labeled data to identify factors that distinguish different labels. Models succesfully leveraging this type of learning are able to label new data.

**Unsupervised** learning is where the model learns from a set of unlabeled data. These models learn structure from the data provided to them and are then able to determine whether and how a new piece of data fits into the learned structure. Unsupervised learning models have the advantage that no prior training is needed. This approach excels at finding data points that are different from others but is unable to easily affix a label to these anomalies or outliers.



Unsupervised learning learns the underlying structure of unlabled data. Models succesfully leveraging this learning are able to measure how well new data fits into a learned structure.

Within these broad approaches are a range of different learning algorithms as shown below, with researchers continuing to create new ones. To complicate things further, algorithms can be combined to form even more complex systems. The question then arises, how does a data scientist choose the right algorithm or algorithms to solve a particular problem? Or can one algorithm be superior to all others no matter what the problem is?

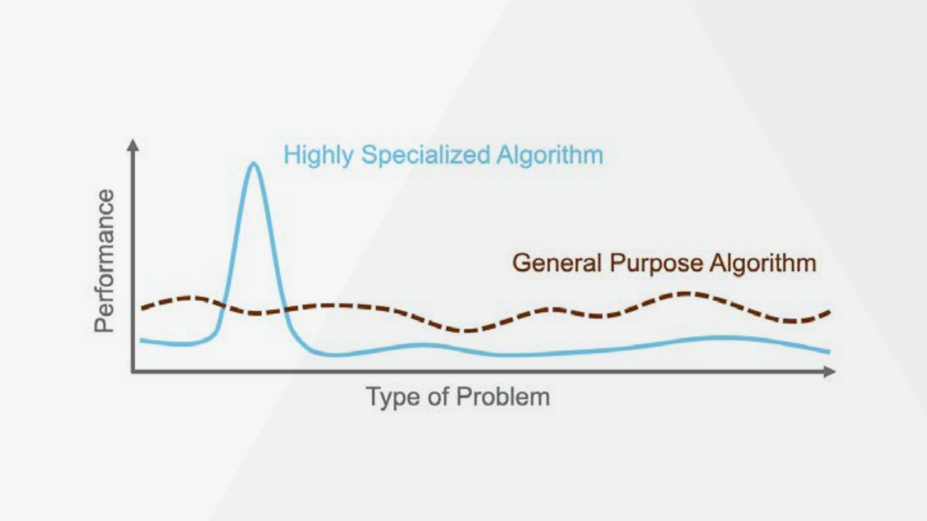


There exist a large number of machine learning algorithms each with different strengths and weaknesses depending on the type of problem being addressed.

## No free lunch theorem

It turns out that there is no single algorithm that will outperform every other one for all possible problem statements. This is called the “no free lunch theorem.” Simply stated, given a problem, there will always be a specialized algorithm that outperforms a generic one for that problem. The need for specific algorithms to address specific problems underpins the need for the ever-growing number of algorithms mentioned above. There are problems where a supervised neural network will perform best and others where unsupervised hierarchical clustering will be best.

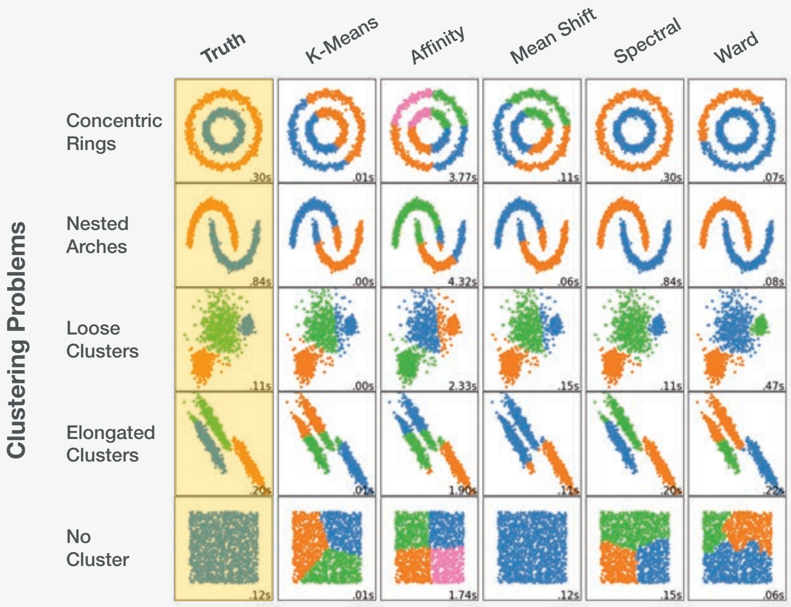
The algorithm used for image recognition in self-driving cars cannot be applied to translate from one language to another. Each algorithm is a specific choice, optimized to the problem that was set out to be solved and the data on which the model operates.



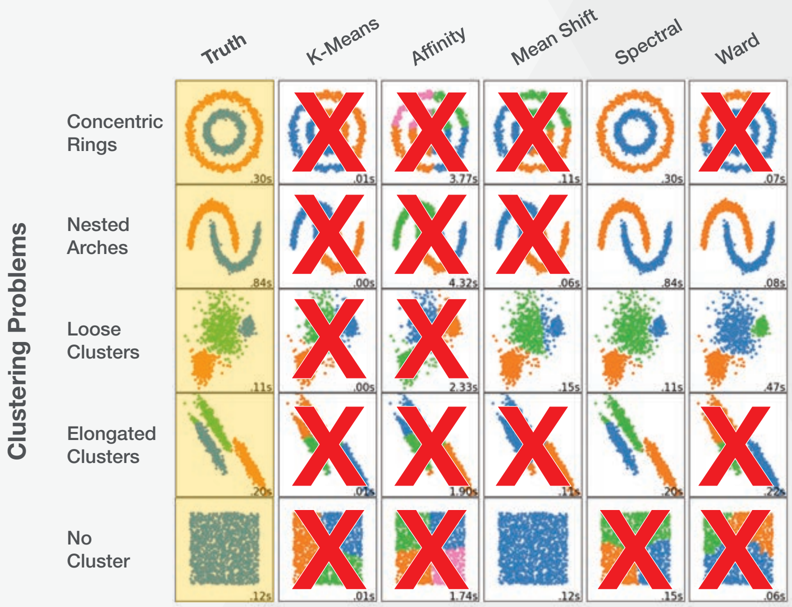
No free lunch theorm: There is no single algorithm that will perform well against every problem.

## Finding the right tool for the job

How does a data scientist then choose the right algorithm? That is as much an art as it is a science. The combination of the problem statement and deep understanding of the data can point the data scientist in the right direction. What is important to realize is if done wrong, the result may not just be sub-optimal, but it may give completely wrong results! Take the example shown below. For each dataset, the choice of algorithm results in widely differing sets of results. There is an optimal choice of algorithm for each problem, but more importantly, certain choices lead to very undesirable results. This makes it extremely important to make sure the right approach is chosen for the right problem.



Comparison of machine learning algorithms (x-axis) results against different datasets (y-axis). True labels highlighted in yellow. Adapted from scikit-learn.org.



Comparison of results. An X is placed on wrong perdictions that would lead to undesirable results. No single algorithm is effective for every dataset. Adapted from scikit-learn.org.

### How Do you Measure *Good*?

An important part of how data scientists choose the right model is deciding how to measure whether a model is successful. Often when talking about model performance the *accuracy* of the model will be referenced.

$$Accuracy = \frac{(True\ Positives + True\ Negatives)}{(True\ Positives + True\ Negatives + False\ Positives + False\ Negatives)}$$

Accuracy as a metric has its value, but a seemingly good accuracy can hide the true story of a model's performance. Consider a classification problem where the goal is to label data with one of two labels, label A or label B. If label A is 1000 times more likely to occur than label B, you can easily achieve 99.9% by always reporting data with label A. While this gives great accuracy you will never correctly label anything as B! Clearly, accuracy is not the right measurement if we care about finding B cases. Luckily, data scientists have additional metrics that help them optimize and measure the model's effectiveness for cases that they care about.

Precision is another such metric. It measures how correct something is at guessing a particular label relative to the total number of guesses of that label that the model makes.

$$Precision = \frac{True\ Positives}{(True\ Positives + \ False\ Positives)}$$

Data scientists that aim for a high precision score will build models that predict labels without generating many false alarms. What precision does not tell us is whether the model failed to label cases that we care about. Recall is another metric that helps provide perspective.

Recall measures how often a model is correct in finding a particular label relative to all instances of that label.

$$Recall \ = \frac{True\ Positives}{(True\ Positives + \ False\ Negatives)}$$

Data scientists that aim for a high recall score will build models that will not fail to alarm on instances that are cared about.

Tracking and balancing both precision and recall allow data scientists to effectively measure and optimize their models for success.

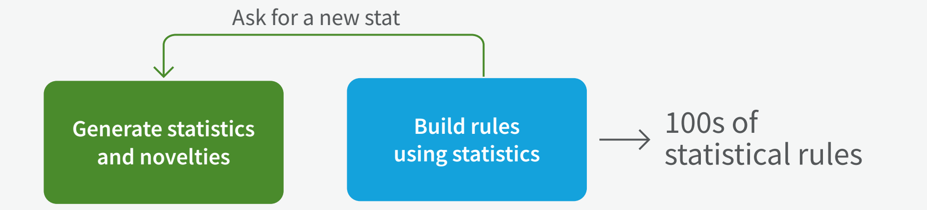
An important part of how data scientists choose the right model is deciding how to measure whether a model is successful.

## Applying AI to threat detection

AI and its many disciplines have a significant role to play in finding and stopping attackers in modern enterprises. Two paradigms have emerged for the active identification of cybersecurity threats – math-led and security-led. In this section, we will break down the differences between these paradigms and explain why security-led AI provides optimal results for security teams.

### Math-led AI – a flawed approach to threat detection

In the math-led paradigm, data scientists generate simple sets of statistics using a limited number of generic algorithms focused on outlier detection or novelty detection. Security researchers then combine these statistics to create hundreds of statistical rules. If a new statistic is needed, the same generic approach is used to create the new one. These statistical rules are often augmented with explicit suppression filters as post-processing to address additional detection volumes that are generated with this type of generic approach (returning to the no free lunch theorem – generic algorithms will result in sub-optimal performance).


Ask for a new stat
Generate statistics and novelties
Build rules using statistics
100s of statistical rules

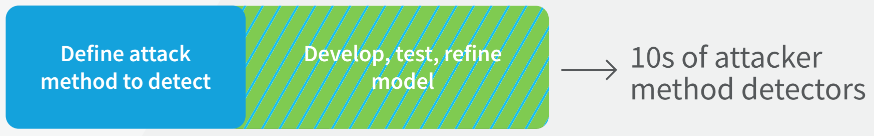
Math-led paradigm for building detections.

As an example, let's try and detect a command-and-control channel. The data science team could start by generating a statistic for the rarity of all external domains. The security research team must then decide on the threshold of rarity that would result in C2 channel detection. If a lot of domains used by IoT devices are above the rare threshold, a suppression filter would need to be applied to ignore all IoT devices. Additional suppression filters on user agents, subnets, and other attributes would be applied until there was a manageable volume of alerts. The generic nature of this approach requires these suppression rules to be present despite the risk they present of blocking an evasion technique used by an attacker.

### Security-led AI – a maximum coverage and minimal noise approach to threat detection

The security-led paradigm is a tightly coupled approach between defining the problem (attacker method) and finding the right model. Security researchers define the problem statement by identifying a broad attacker method, not just a tool or single exploit, and data scientists find the appropriate algorithm to identify that method, working closely with security researchers in iterating over the solution. This approach directly detects the attacker method – and does not just identify cursory anomalies that are often reported in math-led approaches.

The security-led approach to threat detection results in better performance as measured by recall and precision. Beyond these metrics, this approach is resilient to changes in attacker tools and requires fewer detection types resulting in easier operations for security teams. When a new attacker method starts to trend upwards the security-led process starts and a new detection is created. While the sophistication of this approach can require additional development time—attacker methods are very slow to change and consistently appear alongside older and fully covered methods.


Define attack method to detect
Develop, test, refine model
10s of attacker method detectors

Security-led paradigm for building detections.

## How Vectra Works

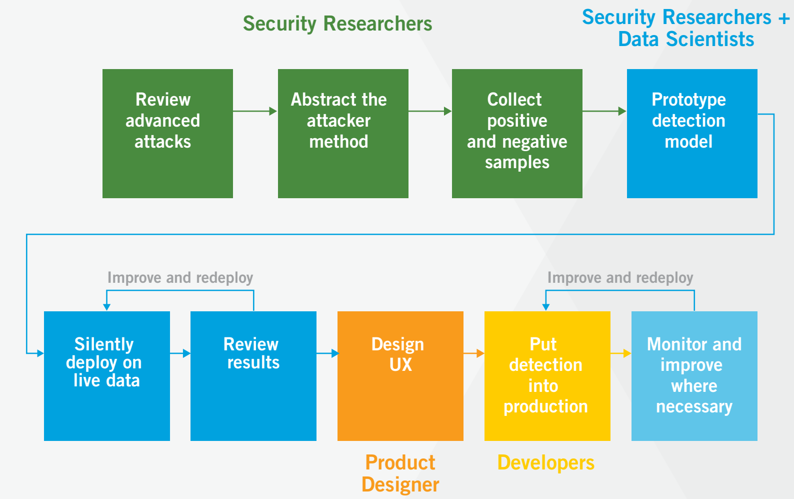
Vectra has pioneered the security-led approach for finding attacker methods in network, public cloud, SaaS apps, and identity. In the following sections, we will dive into the scope of Vectra's coverage and development process, the engine that collects and generates detections, how single events are correlated into actionable security incidents, and the inner workings of two Vectra detections.

### Detection Development at Vectra

Vectra detections focus explicitly on finding attackers and identifying attacker methods in action, not just weird anomalies. Coverage is built by security researchers with diverse backgrounds and data scientists with a deep understanding of how to extract value from massive complex data sets. Over the past ten-plus years, these two groups have developed a highly collaborative approach to threat detection development that scales across security domains and data types to effectively find attacker behaviors with minimal noise.

Vectra's security research team is present throughout the entire detection development process. Their work leads the process, with the team constantly monitoring and reviewing attacker methods being used in the wild. Research is not focused on specific tools or attack groups but on the general methods that attackers are executing. For example, security researchers may see the beacon functionality of Cobalt Strike being used in ransomware incidents.

> Vectra has pioneered the security-led approach for finding attacker methods in network, public cloud, SaaS apps, and identity.


Security Researchers
Security Researchers + Data Scientists
Review advanced attacks
Abstract the attacker method
Collect positive and negative samples
Prototype detection model
Improve and redeploy
Improve and redeploy
Silently deploy on live data
Review results
Design UX
Put detection into production
Monitor and improve where necessary
Product Designer
Developers

How Vectra develops security-led AI algorithms.

Instead of looking only at Cobalt Strike beacons specifically, they abstract the actions of this technology and study the attacker's method of *control*. By focusing on the abstracted method, Vectra can build coverage for both the tools known to execute this method today and the tools that will be developed in the future.

Once the attacker method has been identified, the security researchers work to collect a corpus of malicious and benign samples. Malicious samples are collected from several places including customers who voluntarily share anonymized metadata, synthetic data creation algorithms, publicly documented cyber-incidents, and attacks in our internal labs. Benign samples are harvested from Vectra's large data set of anonymized customer metadata.

With the attacker method and supporting data in hand, the security researchers work with the data science team to develop a prototype model with an optimal threshold for attacker method detection. The prototype is deployed in a silent beta-mode where it runs behind the scenes and reports back summary information from a larger opt-in customer base. To ensure the final model has the highest possible efficacy, the prototype reports back every instance of an attacker method that is observed and every instance of things that look like the attacker method – i.e., events that are just below the threshold. This sub-threshold triggering enables the data scientists the opportunity to further tune their models and ensure no behaviors are missed. The models are rapidly iterated on until strict standards of quality are satisfied for their performance in detecting the attacker methods in the real world.

The final steps of detection development involve the creation of a dedicated UI that shows the full context of the identified attacker method, and where relevant, additional information about what is normal for the systems in question. The models are then deployed into production where they operate and report incidents for customers to view. The same prototype pipeline that is used to collect data is used to monitor the efficacy of the model in the wild and if necessary, used to make additional improvements to the detection.

The results of all this effort are that models do not require frequent tuning and are effective against current and future generations of attacker tools. Vetra's security-led approach excels at detecting attacker actions, not just strange events.

### Real-time Streaming Engine for Actionable Results

Speed of detection matters. Delays in alerting provide attackers the opportunity to progress their attacks further. Vectra algorithms run on streaming data instead of running on periodic batches. This allows Vectra detections to find attackers without delay, ensuring ample time to stop their progression.

The scale of operation matters because the footprint of enterprise networks, cloud deployments, and SaaS services are constantly growing which results in more and more data for Vectra's detections to process. Vectra's real-time streaming engine supports large international enterprises by extracting the necessary data to build long-term learning without issues of data size.

The effectiveness of algorithms, specifically those that use unsupervised learning, are significantly impacted by the amount of history available to them. Running detections in batches limits the amount of data that can ever be processed in a reasonable amount of time. In Vectra's streaming approach, algorithms extract the relevant pieces needed from an event and factor those into new baselines for models. By learning from streaming data, baselines are built off of months of data and millions of events ensuring the highest quality alerts.

### Artificial Intelligence for Threat Correlation

Vectra's AI is not only applied to the identification of individual attacker methods, but also to the correlation of those actions to identify, categorize and prioritize actively progressing attacks. This correlation is necessary because cyber attackers will execute several actions across domains to progress towards a final objective. A dedicated correlation algorithm analyzes behaviors across accounts, hosts, network, and the cloud to present a clear signal of a security incident. The correlation algorithm then attributes the behaviors to stable anchors in the form of accounts or host machines.

For example, in network and hybrid-cloud environments, transient IPs are attributed to stable host machines based on artifacts that are observed via an algorithm called host-id. Artifacts are collected from network metadata which include information like Kerberos host principals, DHCP MAC addresses and cookies—and from API integrations like EDR, vCenter, Azure, and AWS. Once artifacts are attributed to a given host machine, any time an IP is seen with a given artifact, that metadata flow and any associated attacker behavior can be attributed to the named host machine—not just the IP.

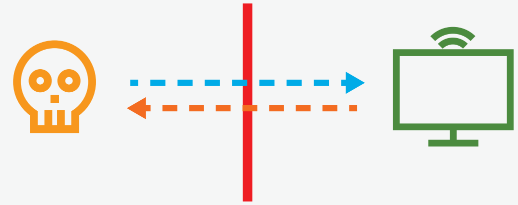
> Vectra's AI is not just applied to the identification of individual attacker methods but also to the correlation of those actions to identify, categorize and prioritize actively progressing attacks.

A different challenge to attribution exists in AWS due to how events are recorded in the AWS control plane, where they are associated with Assumed Roles, not the underlying user accounts. Any number of accounts can assume a given Role, but knowing the actual IAM or SAML user that assumed the Role—is critical for responding to an attack.

Advanced attackers can make this even harder for defenders by chaining roles in an attempt to hide the origin of an attack. Vectra, using a custom-built technology known as Kingpin can step back through the chaining of Roles to attribute observed attacks to an underlying user, not an ambiguous Role.

Once attacker behaviors are attributed to a stable indicator, they are correlated together to identify the underlying behavioral profile of the system which then labels and prioritizes progressing threats. The correlation algorithm was designed to replicate the actions taken by Vectra's analysts and security researchers when investigating threats, to provide the ability to classify advanced attacker scenarios like external threat actors or admin level insider threats for immediate review.

## AI Detection Case Study – Encrypted Command and Control Channels



Attackers use external command and control channels
to control internal systems.

### *Attacker Method*

Core to every network-based attack is the use of a command-and-control channel (C2). Attackers with access to a host machine will deploy malicious software that reaches out to an external server. Despite the internal machine initiating the connection, the external server's responses contain instructions that the infected host machine executes allowing the attacker to progress their attack.

Command and control tools are present in off-the-shelf attack frameworks like Cobalt Strike and Metasploit as well as developed in-house by advanced attack groups. These frameworks all support encryption of the channel as well as other techniques like domain fronting or session jitter to help the attackers evade detection.

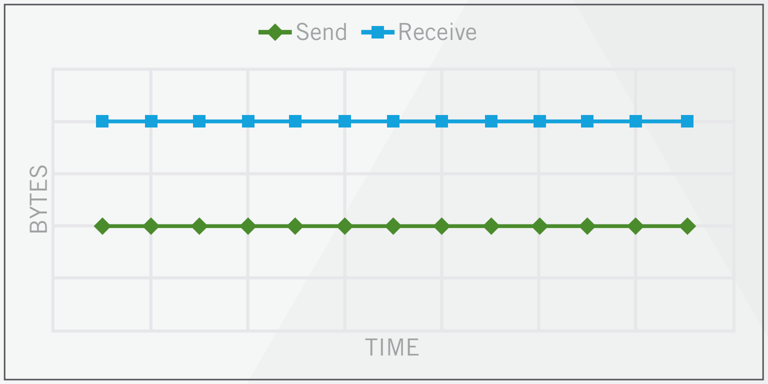
> Vectra detects command and control channels regardless of encryption or other evasion techniques.

### *Detection Methodology*

Vectra detects command and control channels regardless of encryption or other evasion techniques. This coverage comes from the security-led approach described above, addressing many of the issues that come up when attempting to solve the problem with a math-led approach.

When Vectra's security research team abstracted the behavior of a command-and-control channel, they identified that the clearest indicators of the method were not circumstantial aspects of the traffic—it wasn't rare domains or user agents, but instead the actual shape over time of the network traffic.

Consider a representative example of benign traffic from an external system below.

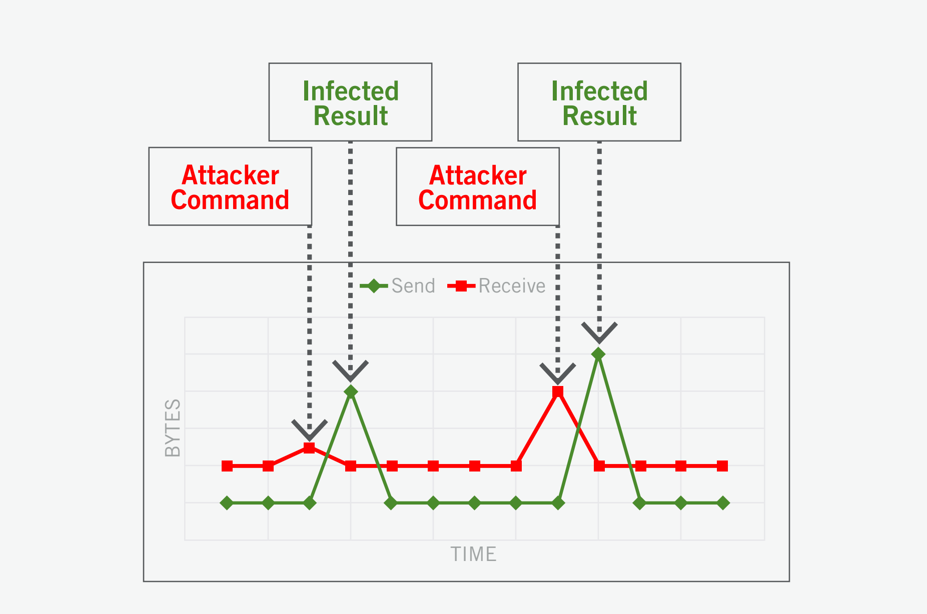


**Benign Traffic**

Example of benign beacon data transfer traffic.

This traffic example characterizes a host machine that is beaconing outbound with an external server. Beacons are a very common network function used by services like stock tickers, chat apps, and ad trackers that enable local and remote systems to remain in sync and communicate. That same functionality is also used by malicious command-and-control channels.

There is a subtle difference however in how a beacon appears when used by a stock ticker and when it is used for a malicious channel. Consider the following data representing a malicious encrypted tunnel.

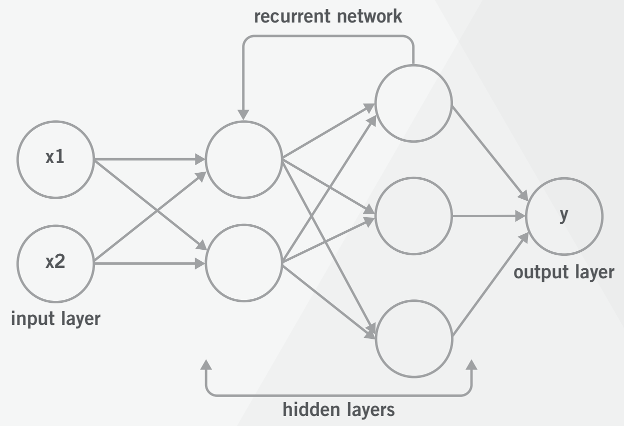


Example of malicious command and control data transfer traffic.

Notice the spike patterns? These occur when the attacker sends their command, and the infected system sends back a result. The first data spike comes unprompted in the "receive bytes" and is quickly followed by the infected machine's response.

Vectra's data scientists studied these patterns and were able to identify an approach that would be optimal at identifying this behavior. The time-series data that characterizes the command-and-control channel behavior has many similarities to the data used in speech recognition and natural language processing, which led the team to decide on the use of a deep learning model.

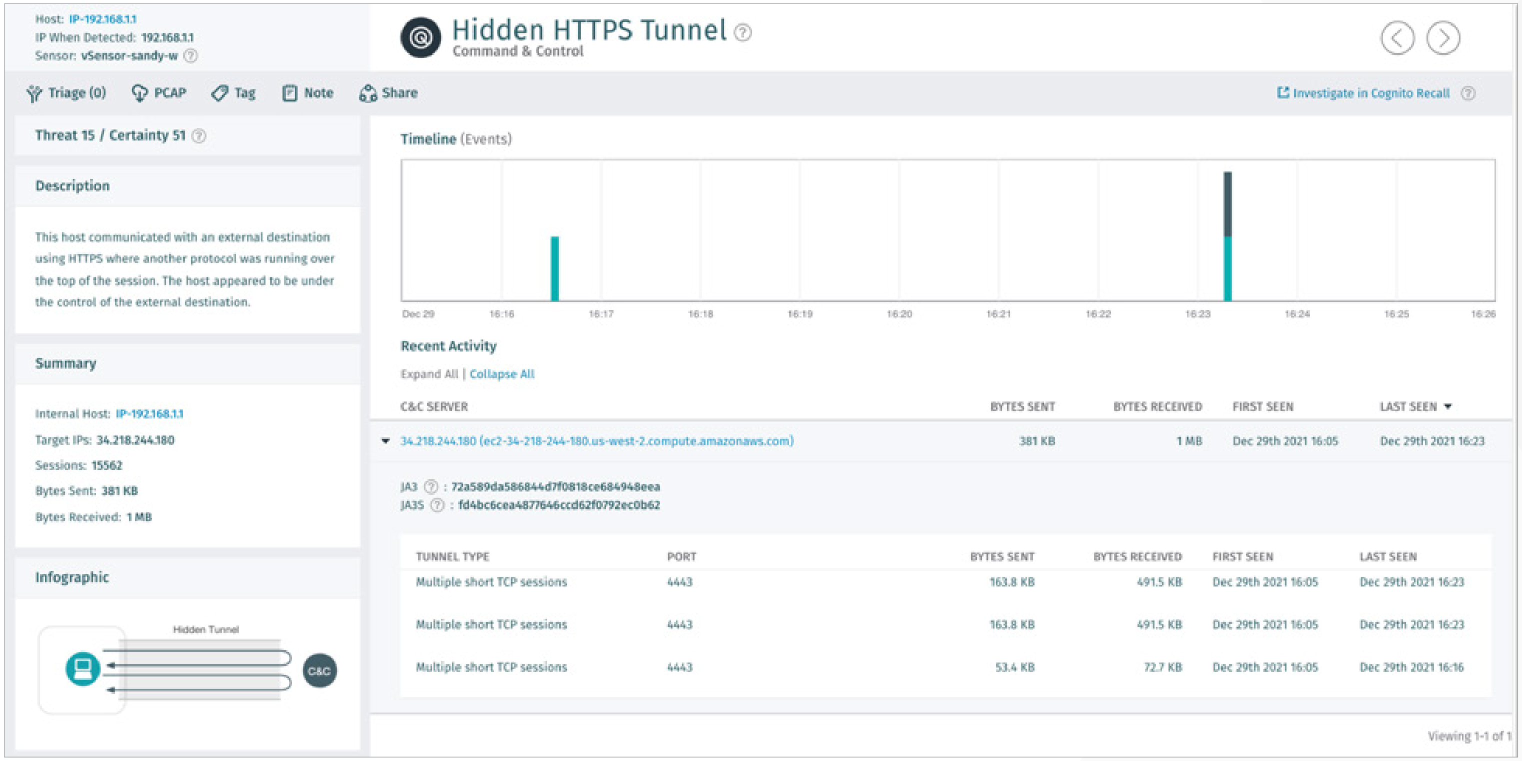
Vectra uses a specific recurrent neural network architecture known as an LSTM (long short-term memory) to identify the attack behavior. This type of algorithm excels at understanding events at multiple different timescales which is key to fully understanding the nature of the command and control conversation data. The LSTM is trained on real and algorithmically generated samples. The data set covers an extensive range of scenarios, tools, configurations, and environments enabling the model to identify the generalizable signal of a control channel regardless of the tool that is used.



Vectra uses recurrent neural networks to differentiate between malicious command and control communication from benign beacons.

It is worth noting as well that this algorithmic approach was made possible by how Vectra formats network session data. While Vectra is capable of outputting Zeek-like metadata, Vectra's custom parser provides additional metadata fidelity beyond that of standard Zeek—with sub-second interval parsing of network communications. This fine-grained view enables clear visibility into all types of benign and malicious communications and allows Vectra's data science teams to use the algorithms that provide the best possible coverage for a wide range of problems.

The result of this unique metadata and the sophisticated algorithmic approach—is the ability to effectively find attackers. The decision to focus on the communication data itself, instead of cursory signals, ensures resiliency against tool changes as well as encrypted traffic. The clear behavior signal also removes the need for suppression filters that could potentially filter fronted channels or stealthy attacker actions.



Host: IP-192.168.1.1
IP When Detected: 192.168.1.1
Sensor: vSensor-sandy-w

Hidden HTTPS Tunnel
Command & Control

Threat 15 / Certainty 51

Description

This host communicated with an external destination using HTTPS where another protocol was running over the top of the session. The host appeared to be under the control of the external destination.

Summary

Internal Host: IP-192.168.1.1
Target IPs: 34.218.244.180
Sessions: 15562
Bytes Sent: 381 KB
Bytes Received: 1 MB

Infographic

Timeline (Events)

Recent Activity

Expand All | Collapse All

| C&C SERVER | BYTES SENT | BYTES RECEIVED | FIRST SEEN | LAST SEEN ▾ |
|---|---|---|---|---|
| ▾ 34.218.244.180 (ec2-34-218-244-180.us-west-2.compute.amazonaws.com) | 381 KB | 1 MB | Dec 29th 2021 16:05 | Dec 29th 2021 16:23 |

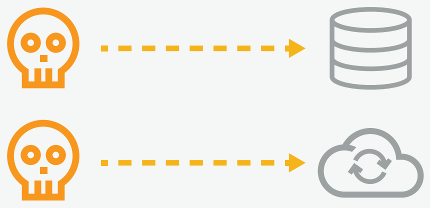
JA3 ⓘ : 72a589da586844d7f0818ce684948eea
JA3S ⓘ : fd4bc6cea4877646ccd62f0792ec0b62

| TUNNEL TYPE | PORT | BYTES SENT | BYTES RECEIVED | FIRST SEEN | LAST SEEN |
|---|---|---|---|---|---|
| Multiple short TCP sessions | 4443 | 163.8 KB | 491.5 KB | Dec 29th 2021 16:05 | Dec 29th 2021 16:23 |
| Multiple short TCP sessions | 4443 | 163.8 KB | 491.5 KB | Dec 29th 2021 16:05 | Dec 29th 2021 16:23 |
| Multiple short TCP sessions | 4443 | 53.4 KB | 72.7 KB | Dec 29th 2021 16:05 | Dec 29th 2021 16:16 |

Viewing 1-1 of 1

Vectra detection for an encrypted command and control channel.

## AI Detection Case Study: Privilege Credential Abuse in the Network and Cloud



Attackers use privilege credentials to move laterally and execute other tactics in the network and cloud.

### *Attacker Method*

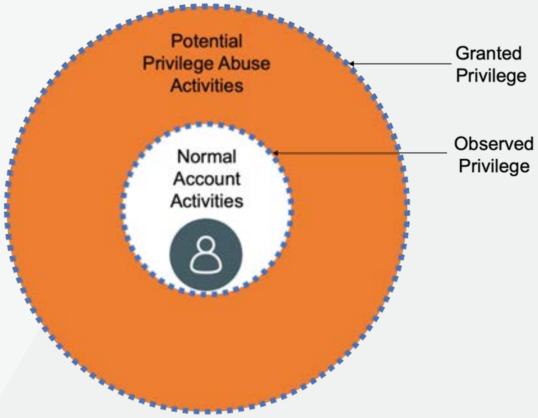
Attackers that obtain privileged credentials gain broad access to network and cloud resources. Credentials allow attacker access without needing to use malware or exploit payloads that can leave trails behind or trigger preventative alarms. Enforcing requirements around the lowest level of privilege for users can help mitigate some threat activity, but recent attack examples demonstrate that this continues to be a challenge.

The challenge of preventing abuse of stolen credential access necessitates detecting when an incident of abuse occurs. The detection of an attacker stealing and leveraging an account presents a unique set of challenges. Every action executed by an attacker is explicitly allowed based on its set permissions. Alerting based on concepts like new or novel interaction will fail to be effective. This is because users exist in dynamic environments where accessing new resources is core to their daily jobs. An attacker who has gained knowledge of the environment will attempt to blend in and perform actions that are not new for a particular account to avoid raising suspicion. To identify privilege credential abuse effectively, a security-led approach is required that considers what an attacker is trying to accomplish with a set of stolen credentials to effectively detect credential abuse.

### *Detection Methodology*

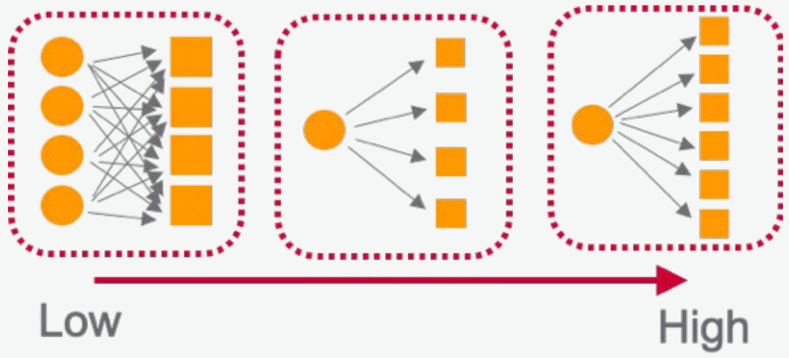
Vectra can identify the abuse of stolen privilege credentials in both network and cloud environments. Core to this security-led detection approach is an understanding of what attackers do with stolen credentials. The value of privileged credentials to an attacker is the ability to access services and functionality regarded as high value and privileged in the environment.

Vectra's security researchers identified that if you knew the actual privilege of every account, host machine, service, and cloud operation—you would have a map of all the high-value resources that exist. While concepts of *granted privilege* are well established, this representation provides an upper bound to what the true privilege of something is compared to the minimum necessary privilege. Instead, Vectra's security research team and data science team identified a new way of representing the value of systems in an environment based on what was observed over time. This dynamic and ground view of value is called *observed privilege*. This data based view of privilege provides an effective zero-trust approach to credential use without manual configurations.


Potential Privilege Abuse Activities
Normal Account Activities
Granted Privilege
Observed Privilege

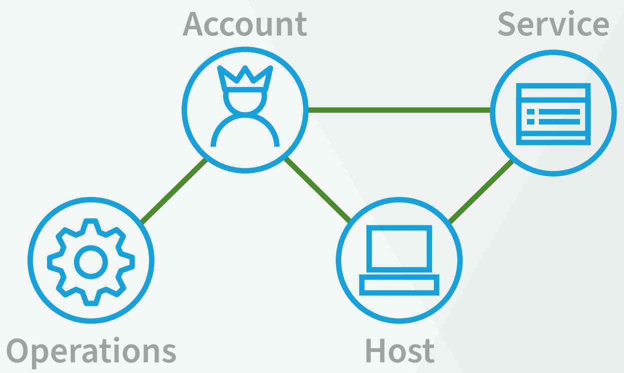
Observed privilege is a zero-trust view of the normal privilege a user needs to do their job. Use of privilege beyond what is normally necessary warrants additional scrutiny.

Vectra's AI calculates the *observed privilege* by considering the historic interactions between the tracked entities, not the privilege that is defined by an IT admin. The breadth and specificity of access and usage heavily contribute to the scores. A system that accesses several systems that are normally accessed by other systems will have a low privilege whereas a system that accesses a high number of systems that are not accessed by others will have a high privilege score. This approach allows Vectra to differentiate between domain admin accounts and normal user accounts.



Vectra learns observed privilege levels based on user behavior. An account that accesses a lot of common services has lower privilege than one that accesses services few others access.

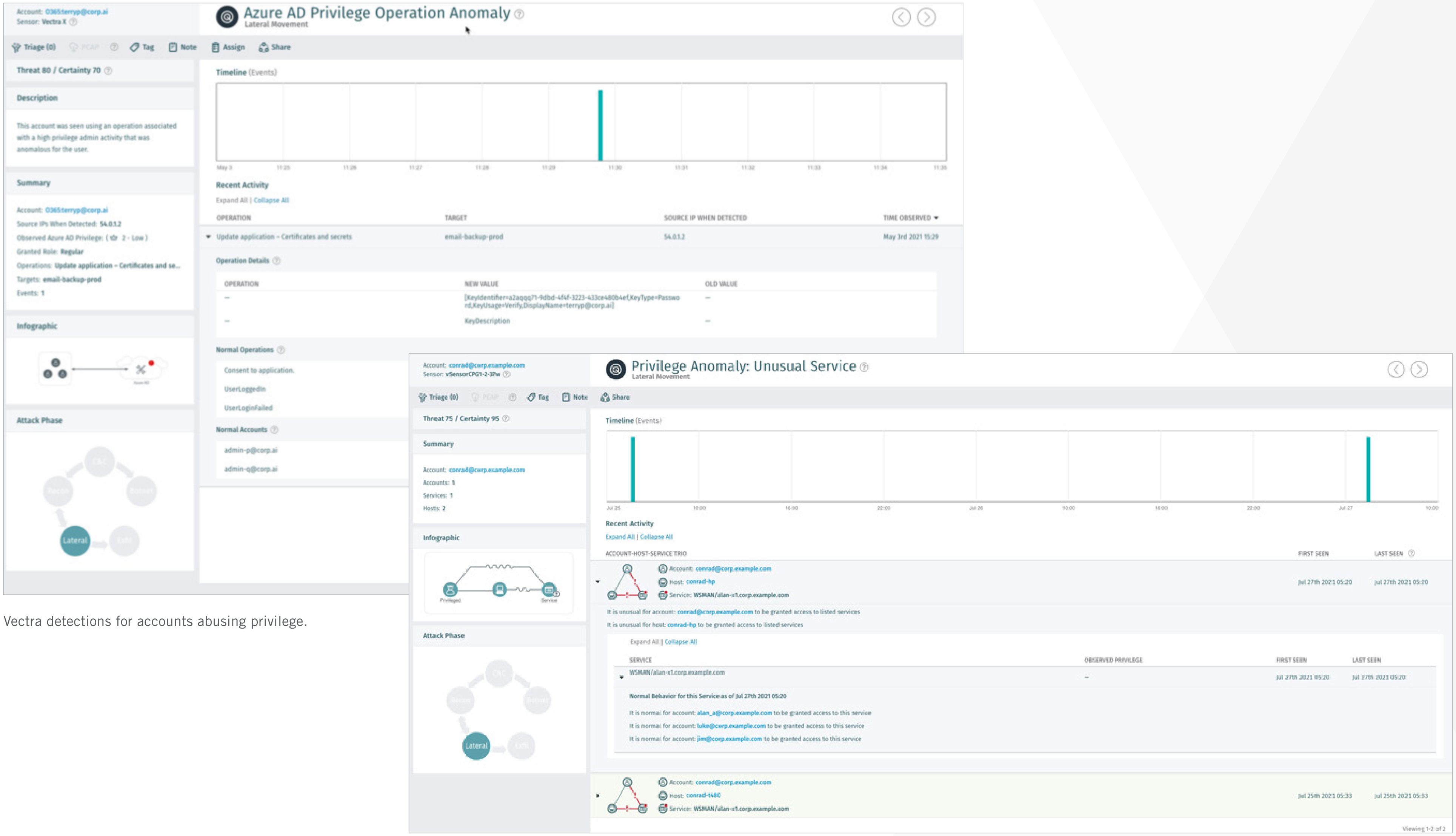
Once observed privilege scores have been calculated, all the interactions between accounts, services, hosts, and cloud operations are mapped to understand the normal historical interactions between systems. Then, a suite of unsupervised learning algorithms that consider the privilege scores identify anomalous cases of privilege abuse, where custom anomaly detection algorithms and implementations of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) are used.



Vectra applies unsupervised learning that considers observed privilege and the interactions between accounts, hosts, services and cloud operations in order to find credential abuse.

The results of this sophisticated security-led approach are the ability to identify stolen credentials that are abused in both the cloud and in on-premises networks. The *observed privilege* metric focuses the detection on the anomalous actions that matter and enables both higher precision and recall than an approach that ignores this critical perspective.

Vectra's AI calculates the *observed privilege* by considering the historic interactions between the tracked entities, not the privilege that is defined by an IT admin.



Account: O365:terryp@corp.ai
Sensor: Vectra X

## Azure AD Privilege Operation Anomaly

Lateral Movement

Threat 80 / Certainty 70

Description

This account was seen using an operation associated with a high privilege admin activity that was anomalous for the user.

Summary

Account: O365:terryp@corp.ai
Source IPs When Detected: 54.0.1.2
Observed Azure AD Privilege: ( ☆ 2 - Low )
Granted Role: Regular
Operations: Update application – Certificates and se...
Targets: email-backup-prod
Events: 1

Timeline (Events)

Recent Activity

| OPERATION | TARGET | SOURCE IP WHEN DETECTED | TIME OBSERVED ▾ |
|---|---|---|---|
| ▾ Update application – Certificates and secrets | email-backup-prod | 54.0.1.2 | May 3rd 2021 15:29 |

Operation Details

| OPERATION | NEW VALUE | OLD VALUE |
|---|---|---|
| – | [KeyIdentifier=a2aqqq71-9dbd-4f4f-3223-433ce480b4ef,KeyType=Password,KeyUsage=Verify,DisplayName=terryp@corp.ai] | – |
| – | KeyDescription | – |

Normal Operations

Consent to application.
UserLoggedIn
UserLoginFailed

Normal Accounts

admin-p@corp.ai
admin-q@corp.ai

Account: conrad@corp.example.com
Sensor: vSensorCPG1-2-37w

## Privilege Anomaly: Unusual Service

Lateral Movement

Threat 75 / Certainty 95

Summary

Account: conrad@corp.example.com
Accounts: 1
Services: 1
Hosts: 2

Timeline (Events)

Recent Activity

| ACCOUNT-HOST-SERVICE TRIO | FIRST SEEN | LAST SEEN |
|---|---|---|
| ▾ Account: conrad@corp.example.com<br>Host: conrad-hp<br>Service: WSMAN/alan-x1.corp.example.com | Jul 27th 2021 05:20 | Jul 27th 2021 05:20 |

It is unusual for account: conrad@corp.example.com to be granted access to listed services
It is unusual for host: conrad-hp to be granted access to listed services

| SERVICE | OBSERVED PRIVILEGE | FIRST SEEN | LAST SEEN |
|---|---|---|---|
| ▾ WSMAN/alan-x1.corp.example.com | – | Jul 27th 2021 05:20 | Jul 27th 2021 05:20 |

Normal Behavior for this Service as of Jul 27th 2021 05:20

It is normal for account: alan_a@corp.example.com to be granted access to this service
It is normal for account: luke@corp.example.com to be granted access to this service
It is normal for account: jim@corp.example.com to be granted access to this service

| ACCOUNT-HOST-SERVICE TRIO | FIRST SEEN | LAST SEEN |
|---|---|---|
| ▸ Account: conrad@corp.example.com<br>Host: conrad-t480<br>Service: WSMAN/alan-x1.corp.example.com | Jul 25th 2021 05:33 | Jul 25th 2021 05:33 |

Viewing 1-2 of 2

Vectra detections for accounts abusing privilege.

## Out Innovate and Out Work Attackers

Attackers will continue to innovate, which is why defenders must do the same. Vectra has continuously innovated over the years to develop the most effective platform possible for threat detection and response for the on-premises and cloud assets.

Vectra has developed 100+ security-led AI detections and identified countless threats in customer networks and cloud environments, resulting in attackers being stopped from reaching their objective. Each detection was built with a deep understanding about how attackers conduct attacks, with some of the most advanced ML techniques available. In total Vectra has 33 patents for the technology that support these detections.

Beyond the coverage that Vectra's patented technology provides, we are proud to be the most referenced vendor in NSA and MITRE's framework that defines the countermeasures defenders need to protect their environment known as MITRE D3FEND. The D3FEND framework maps how defenders can stop attacks and counter the attacker techniques defined in MITRE's ATT&CK framework. In total, the D3FEND framework references 12 different Vectra patents, which are used as references for defender countermeasures.

We at Vectra are committed to making the world a safer and fairer place. As such we will continue to leverage security-led AI to innovate and build detection capabilities that stop attackers from achieving their goals.

**For more information please contact us at info@vectra.ai.**

Email info@vectra.ai vectra.ai