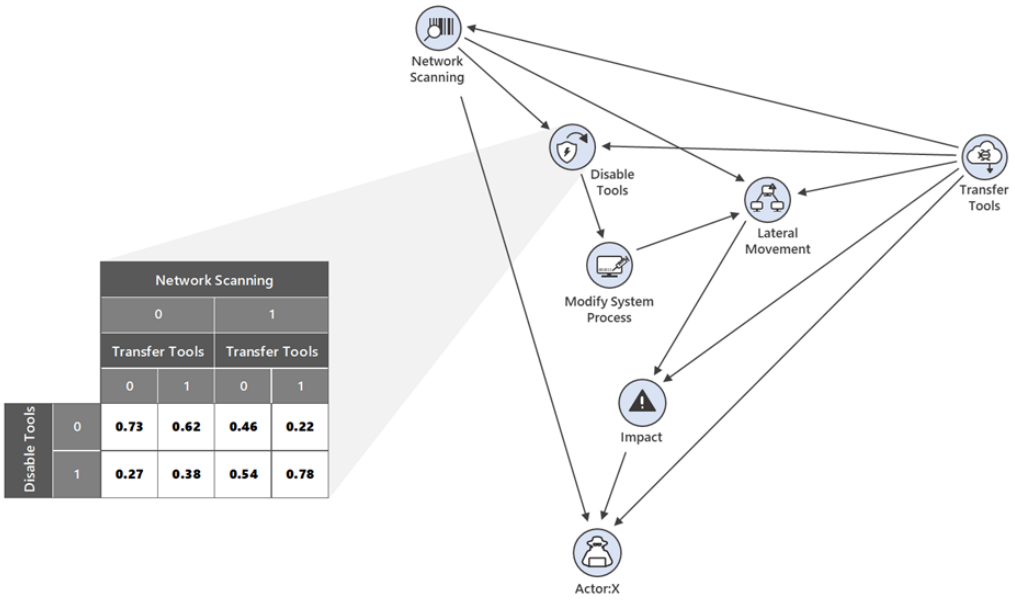
Automating threat actor tracking: Understanding attacker behavior for intelligence and contextual alerting

microsoft.com/security/blog/2021/04/01/automating-threat-actor-tracking-understanding-attacker-behavior-for-intelligence-and-contextual-alerting/

April 1, 2021



***Update [5/9/2022]**: In line with the recently announced expansion into a new service category called **Microsoft Security Experts**, we're introducing the availability of Microsoft Defender Experts for Hunting for public preview. Defender Experts for Hunting is for customers who have a robust security operations center but want Microsoft to help them proactively hunt for threats across Microsoft Defender data, including endpoints, Office 365, cloud applications, and identity.*

As seen in recent sophisticated cyberattacks, especially human-operated campaigns, it's critical to not only detect an attack as early as possible but also to rapidly determine the scope of the compromise and predict how it will progress. How an attack proceeds depends on the attacker's goals and the set of tactics, techniques, and procedures (TTPs) that they utilize to achieve these goals. Hence, quickly associating observed behaviors and characteristics to threat actors provides important insights that can empower organizations to better respond to attacks.

At Microsoft, we use statistical methods to improve our ability to track specific threat actors and the TTPs associated with them. Threat actor tracking is a constant arms race: as defenders implement new detection and mitigation methods, attackers are quick to modify

techniques and behaviors to evade detection or attribution. Manually mapping specific indicators like files, IP addresses, or known techniques to threat actors and keeping track of changes over time isn't effective or scalable.

To tackle this challenge, we built probabilistic models that enable us to quickly predict the likely threat group responsible for an attack, as well as the likely next attack stages. With these models, security analysts can move from a manual method of investigating small sets of disparate signals to probabilistic determinations of likely threat groups based on all activity observed, comparing the activity against all known behaviors, both past and present, encoded in the model. These models help threat intelligence teams stay current on threat actor activity and help analysts quickly identify behaviors they need to analyze when investigating an attack.

In this blog we'll outline a probabilistic graphical modeling framework used by Microsoft 365 Defender research and intelligence teams for threat actor tracking. Microsoft Threat Experts, our managed threat hunting service, utilizes this model to enhance our ability to quickly notify customers about attacks in their environments through targeted attack notifications. These notifications provide technical information and remediation guidance designed to empower customers to identify and mitigate critical threats in their environments.

The model enriches targeted attack notifications with additional context on the threat, the likely attacker and their motivation, the steps the said attacker is likely to make next, and the immediate action the customer can take to contain and remediate the attack. Below we discuss an incident in which automated threat actor tracking translated to real-world protection against a human-operated ransomware attack.

Predicting human-operated ransomware groups

The probabilistic model we discuss in this blog aids Microsoft Threat Experts analysts in sending quick, context-rich, threat actor-attributed notification to customers in the earliest stages of attacks. In one recent case, for example, the model surfaced high-confidence data indicating initial stages of a new ransomware actor in an organization just two minutes into the attack. This enabled analysts to quickly confirm the malicious behavior and the involved threat group, then send a targeted attack notification to the customer, who was able stop the threat before attackers can encrypt data and ask for ransom:

1. The attacker compromises a device via Remote Desktop. This signal, one of many, starts the examination of the attack by the model, which knows that initial access via Remote Desktop is a technique often utilized by a certain threat actor.
2. Attackers copy common open-source tools and custom payloads to the device for such malicious activities as tampering with AV and credential theft, which would allow discovery and lateral movement. With these tools on the device, the model's confidence increases.

3. The attacker begins running the tools and exhibiting behaviors typically associated with attacks by the threat actor.
4. Just two minutes into the attack, the model hits a threshold for activity that indicates the suspected threat actor is present in the organization.
5. Microsoft Threat Experts analysts are notified of the suspected actor activity identified by model, and they quickly send a high-context targeted attack notification that includes technical information as well as actor attribution.
6. As the attacker was attempting to tamper with the antivirus solution, the organization stops the attack, armed with the knowledge of the likely forthcoming activity they need to stop. The threat actor is stopped from performing their other known TTPs, ultimately preventing the ransomware deployment and activation.

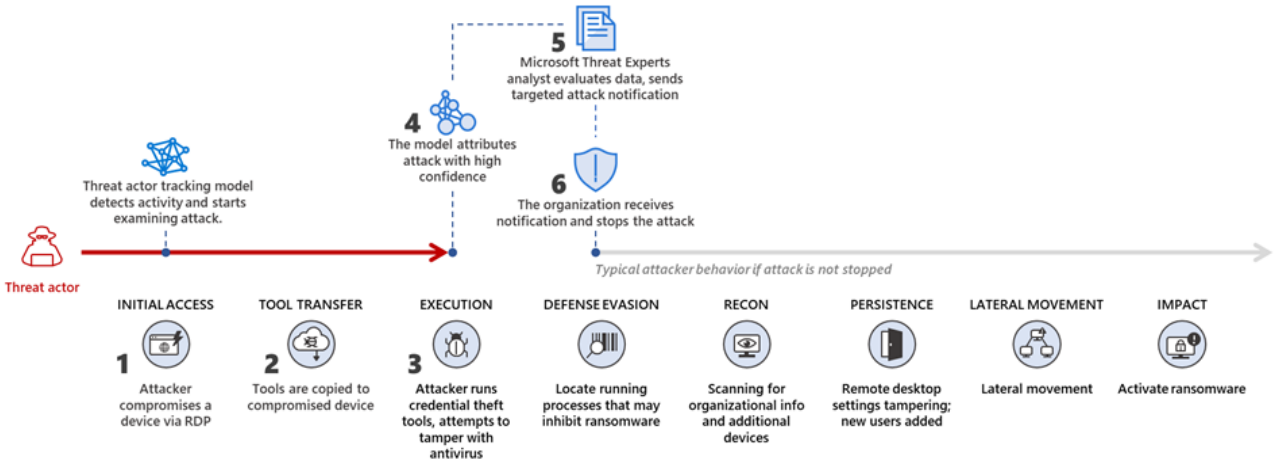


Figure 1. Model predicting human-operated ransomware attack chain

Through the automated threat actor tracking model, Microsoft Threat Experts analysts were able to equip the organization with information about the attack as it was unfolding. The model-enriched targeted attack notification enabled the customer to stop a known human-operated ransomware group before they could cause significant damage. If not stopped, the threat actor would have been able to perform its typical behaviors, including clearing of event logs, creating a persistence method, disabling and deleting backups and recovery options for the device, and encryption and ransom.

Threat actor tracking through probabilistic graphical modeling

As the case study above shows, the ability to identify attacks with high confidence in the early stages is improved by rapidly associating malicious behaviors with threat actors. Using a probabilistic model to predict the likely threat actor behind an attack removes the need for analysts to manually evaluate and compare techniques and tools with known behaviors with threat groups.

Even with attackers frequently adjusting their toolkits, payloads, and techniques to evade detection, the model can help analysts learn new TTPs and then rapidly evaluate the behaviors to confirm the model's prediction. This intelligence allows pivoting to find recently created attacker infrastructure and tools, and increases the ability to report, detect, slow, and stop the adversary.

In the next sections, we will provide more detail about this automated threat actor tracking model and discuss challenges, such as data collection and tagging. We will also share how we leverage security analyst expertise to continuously enrich these models with newfound attacker behavior and improve its ability to surface incidents with high confidence.

Data collection

The first challenge in threat prediction is translating data collected from recorded attacks into a set of well-defined TTPs. The idea is to define a knowledge base such that the approach is generalizable across different threat actor groups. For this purpose, we use the MITRE ATT&CK framework, which provides such a knowledge base and is widely used across the industry for classifying attack behaviors and understanding the lifecycle of an attack.

Attack behaviors need to be carefully mapped at the right level of granularity. If the behaviors are mapped to too broad a category (e.g., MITRE ATT&CK techniques like lateral movement), then discrete attackers cannot be distinguished. If the attack behaviors are too specific (e.g., documented adversary use of a specific file hash) any subtle changes to the behavior or tools used for a particular attack could be missed.

The model uses threat data from Microsoft Defender for Endpoint, as well as the broader Microsoft 365 Defender, which delivers unparalleled cross-domain visibility into attacks. Incidents, which are collections of alerts related to a specific attack, that have been tagged as associated with a threat group correspond to a training sample. These incidents are augmented with more specific indicators of compromise, custom behavioral detections built by our threat hunting teams, and additional context from telemetry. This collection of alerts and detections are then mapped to the collection of TTPs being tracked.

The TTPs are used as variables in a Bayesian network model, which is a statistical model well suited for handling the challenges of our specific problem, including high dimensionality, interdependencies between TTPs, and missing or uncertain data.

Bayesian networks

Given TTPs of an attack observed in an organization, the goal is to identify the most likely threat actor involved and, consequently, the next attack stages, considering that any one TTP very rarely provides enough evidence to attribute an attack to a threat group. It's the combination of these TTPs that provides the necessary evidence to identify the threat group.

We use Bayesian networks to model the relationship of TTPs and threat groups. Bayesian networks are a powerful tool that builds a joint distribution over a set of variables and encodes the relationship between them, which can be represented as a directed acyclic graph. Bayesian networks have properties that make them well-suited for this problem. For one, they are ideal for querying probabilities for a subset of unobserved variables (e.g., attacker groups) in the presence of other observed variables (TTPs). They are also ideal for handling missing or sparse data. Finally, using Bayesian models provides a principled approach to encoding expert knowledge through prior probability distributions that encode one's belief about the quantity of interest before data is considered. With these properties, Bayesian networks have been shown to work well in correlating alerts from various detection systems and predicting future attack stages.[i] [ii]

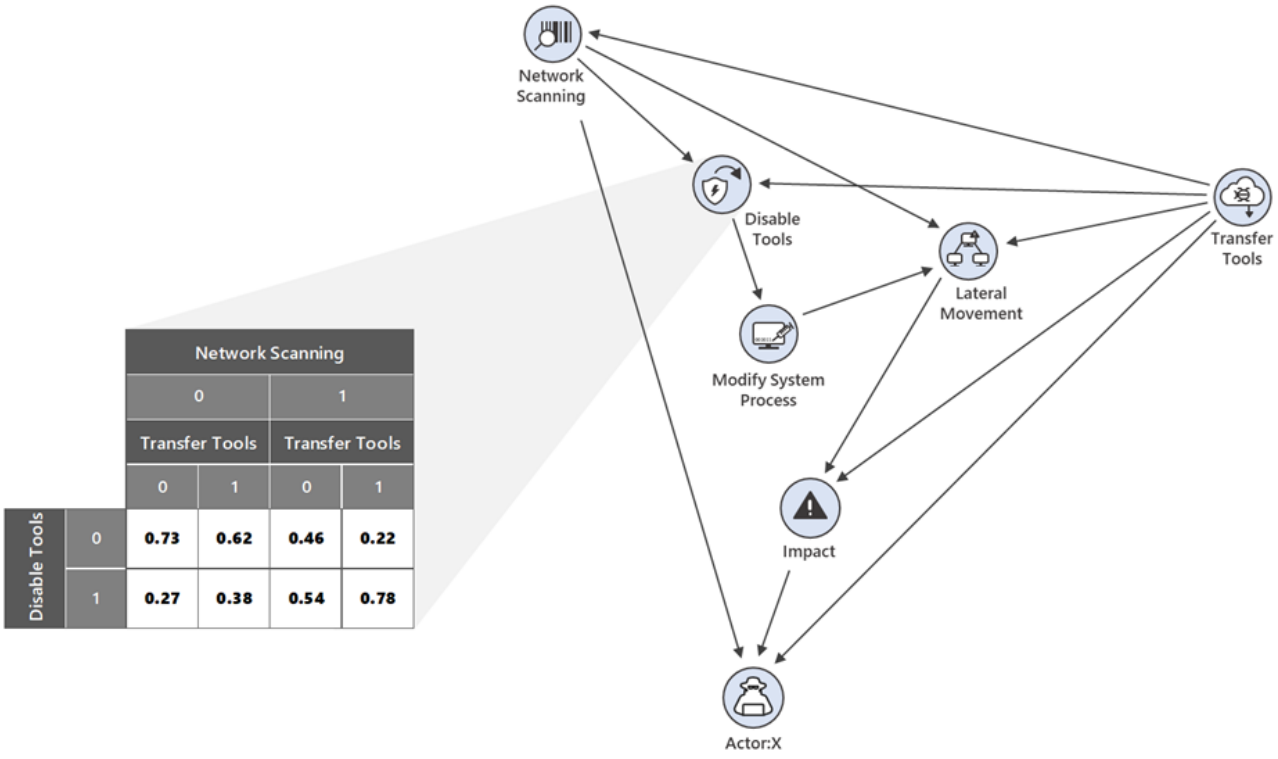
More formally, the set of possible TTPs for an actor are viewed as discrete random variables. Let $\mathbf{X = \{X_1, \ldots, X_n\}}$, where each variable can take on one of two states, 0 or 1. The value of 1 corresponds to the TTP having been observed. Let the random variable **Y** correspond to the indicator variable for a specific threat actor or group of threat actors. Each variable is a node in a directed acyclic graph and the edges between the nodes encode the conditional dependencies between them.

A Bayesian network defines a joint distribution over the set of TTPs and threat actor group, so that:

$$\mathbf{P(X_1, \ldots, X_n, Y) = P(Y|Pa(Y)) \prod_{j=1\ldots n} P(X_i|Pa(X_i))},$$

where $\mathbf{P(X_1, \ldots, X_n, Y)}$ denotes the joint probability of the variables and threat actor group taking on specific values, $\mathbf{P(X_i)}$ denotes the set of parents of variable $\mathbf{X_i}$ in the graph, and $\mathbf{P(X_i|Pa(X_i))}$ the probability that variable $\mathbf{X_i}$ takes on a certain value given (represented by **|**) the state of its parents in the graph. The conditional probabilities of observing a node being 0 or 1 given the set of parent states are represented by conditional probability tables.

Figure 2 shows a toy example where the variable Actor:X corresponds to the threat actor group, with six TTPs inspired by the MITRE ATT&CK framework, including T1570 (Lateral Tool Transfer), T1046 (Network Service Scanning), T1021 (Remote Services), T1562.001 (Impair Defenses: Disable or Modify Tools), T1543 (Create or Modify System Process), and Impact (TA0040; in this example, we do not specify the sub-technique, though that could easily be done). To illustrate, a directed edge between Transfer Tools and Actor:X indicates that the likelihood of observing the actor is directly related to whether we saw them transfer their attack tools. The node Disable Tools shows an example of a conditional probability table and how the probability of observing the technique changes with respect to the states of its parent nodes in the graph, Network Scanning and Transfer Tools.



		Network Scanning			
		0		1	
		Transfer Tools		Transfer Tools	
		0	1	0	1
Disable Tools	0	**0.73**	**0.62**	**0.46**	**0.22**
	1	**0.27**	**0.38**	**0.54**	**0.78**

Figure 2: A toy example showing a Bayesian network for Actor:X with six TTPs. A conditional probability table is also shown for variable Disable Security.

There are two inference tasks that are needed to fully specify the Bayesian network:

1. Structure learning: Given a set of training examples, estimate the graph that captures the dependencies between the variables.
2. Parameter learning: Given a set of training examples and the graph structure, learn the unknown parameters for the conditional probability tables $\mathbf{P(X_i|Pa(X_i))}$.

Structure learning is largely driven by domain knowledge and eliciting expert feedback, which is covered in the next section. Parameter learning is done in the usual Bayesian way, where a prior distribution is specified for the unknown parameters, which can encode subject matter expertise. Then, the parameters are updated with data or new incidents as they arise, so that the final posterior probabilities reflect the prior beliefs from threat intelligence analysts and relevant evidence seen in the data. As new training data is obtained over time as part of hunting and investigations, the Bayesian network can easily be updated so that it always reflects the latest information on the threat actor TTPs.

Because the Bayesian network defines a complete model for the variables and their relationships, it allows the analysts to query for information about any subset of variables and receive probabilistic responses. For example:

- Given Transfer of Tools and Disable Security Tools have been observed but not Modify System Process, what is the topmost likely set of TTPs that will be observed next?

- Given Lateral Movement has been observed, what is the likelihood of seeing Impact?
- Given Network Scanning and Modify System Process, what is the probability that it is threat actor group Actor:X?

This model is particularly useful for its ability to *marginalize* over unobserved variables. For example, if one does not have enough confidence to say whether Impact occurred or not, one can sum over all possible states for that variable and still be able to answer any of the questions above, providing a probabilistic response that reflects that uncertainty.

Finally, the interpretability of these graphical models is high. Analysts can readily see how observing certain techniques directly changes the probability of observing a threat actor or other techniques through the conditional probability tables. In addition, the graph allows easy visualization of how the techniques relate to each other and influence the variable representing the threat actor group.

Threat intelligence elicitation

The combination of minimal training examples with the high dimensionality of the set of possible techniques makes it critical to leverage domain knowledge and threat intelligence expertise.

Our statisticians work closely with threats analysts to incorporate the analysts' large existing knowledge base into the model. Analysts help with learning the structure of the Bayesian network by informing which nodes are likely a-priori to be correlated with each other. For instance, analysts might suggest that they often see Network Scanning followed by Lateral Movement. As we are largely concerned with post-breach attacks, the attack chain defines an inherent sequence of stages that are observed as an attacks progress, such as moving from gaining access to exploitation. This sequencing can help inform the orientation of the edges. Any remaining possible edges are learned from the training examples using one of the structure learning algorithms.[iii]

Once the attack graph is fully specified, the threat analysts help inform the strength of the relationships between the nodes (e.g., how much more likely it is to see Disabling Security Tools given Transfer Tools); this data is encoded in the prior to complete the specification of the model.

Finally, as a threat group changes their behavior over time, new nodes corresponding to new TTPs may need to be added or removed from the graph. This can be done by setting priors based on information from threat intelligence experts and using the alert database to assess correlations with other techniques already in the graph.

Figure 3 illustrates the expert-augmented probabilistic graphical modeling framework. Applying probabilistic learning over these constructed graphs, built from both data collected from real attacks and the vast knowledge of the threat intelligence community, provides a

framework for both predicting the likely threat actor and predicting how an attack might evolve.

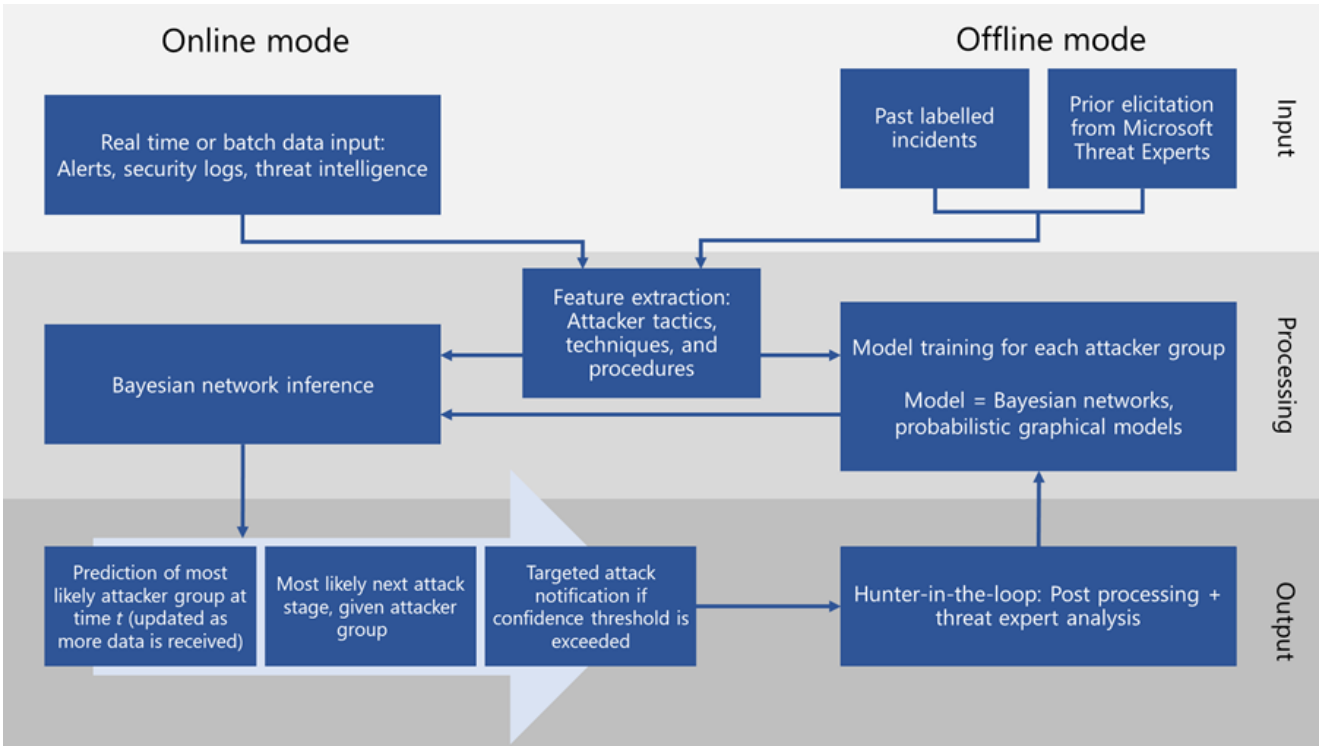


Figure 3. Sketch of framework

Conclusion

Across Microsoft, we use statistical models and machine learning to uncover threats hidden in billions of low-fidelity signals. The threat actor tracking model we introduced in this blog is exciting work with real impact in customer protection. We are still in the early stages of realizing the value of this approach, yet we already have had much success, especially in detecting and informing customers about human-operated attacks, which are some of the most prevalent and impactful threats today.

A core reason for this success is the combination of statistical expertise, threat hunting, and the very intensive work of vetting and discovering the combination of TTPs that indicate specific threat groups. Our ability to automatically identify threat actors from the data, predict next steps, and stop attacks is foundational for much of our work going forward, with many as-yet unrealized benefits in customer protection. In real terms, we have accelerated threat hunting to drive to conclusions that lead to real protection, and we will continue expanding that protection for our customers through the Microsoft Threat Experts service and the coordinated defense delivered by Microsoft 365 Defender.

Cole Sodja, Justin Carroll, Melissa Turcotte, Joshua Neil

Microsoft 365 Defender Research Team

[i] Attack plan recognition and prediction using causal networks

[ii] Real time alert correlation and prediction using Bayesian networks

[iii] A Tutorial on Learning With Bayesian Networks