kaspersky

Historical data analysis in security operations: the role of retrospective search

This essay was written by Sergey Soldatov, Kaspersky's Head of Security Operations Center. It explores the unique role of threat hunting in detecting advanced persistent threats (APTs) that evade automated security solutions, positioning it as a critical component of a modern SOC's detection and response strategy. Drawing from real-world detection practices, it outlines how threat hunting complements alert-driven SOC operations through retrospective analysis and hypothesis-driven investigation, using telemetry data such as EDR/NDR logs.

Threat hunting is the only way to find advanced persistent threats (APTs) because it can unearth threats that are not detected and remediated by automatic detection solutions. APTs are carefully planned, and attackers understand the security controls they must evade to go undetected. Threat hunting therefore covers the gap between offensive activities and automatic security solutions' detection capabilities. It is conducted assuming the monitored infrastructure has been compromised and attempts to prove that compromise. This assumption of compromise means existing security solutions were evaded, and so the only way to find the breach is manual log analysis. An analyst needs detailed logs of activities on operating systems (OS) and networks, and these EDR/NDR-like logs are known as "telemetry," which is the raw data for detection logic and the base for manual threat hunting.

Threat hunting is similar to threat research, wherein an expert tries to spot suspicious activity using telemetry from internet sensors. They then investigate to ensure it's a genuine threat and provide detection logic. This is essentially how malware research works. APTs, however, are unique for each victim. They are impossible to find in the internet and exist only in the victim's infrastructure. Threat hunting is operationalized threat research that's localized for a particular infrastructure; it enables the discovery of advanced threats that have evaded security solutions.

But there is always a trade-off in security, and detection engineering is no exception: higher detection-logic sensitivity means more false positives (FPs) for a SOC team to analyze. That's why, in practice, we divide the detection logic depending on the accuracy, i.e., depending on detection rules' true positive rate (TPR). Internally, we have different types of detection logic. This can be based on atomic behavior patterns in telemetry, such as process start or network connections. It can also involve a complex sequence of atomic telemetry events, like the start of a process with unknown reputation after a successful network logon preceded by several unsuccessful attempts.

We often say that detection logic "hunts" malicious behavior. The official term is "indicators of attack" (IoAs), but we'll call them "detectors" for clarity's sake. TPR is the characteristic of a detector's accuracy – the ratio of the detector's true positives to the sum of its detections (true positives plus false positives). Detectors with a high TPR (and thus a low number of FPs) may be converted into alerts that are processed by the SOC team. Detectors that often generate FPs – but are still an important sign of offensive activity – are left as enrichment for suspicious events.

Earlier we noted that threat hunting is conducted under the assumption of breach, which means it's performed as a retrospective search on historical data. In practice, it's executed as a dedicated internal process within the SOC, usually by tier two analysts. Tier one analysts are responsible for operative alert processing within a predefined alert processing time, typically specified in a service level agreement. To conduct retrospective threat hunting, we must understand what an attack or suspicious activity looks like – and what traces may be left in collected telemetry. Bearing in mind possible traces, the SOC team hypothesizes what may be considered an IoC or IoA. In practice, we have numerous possible hypotheses – a whole hypothesis database where they are classified based on their TPR, use scenarios and other parameters. The aforementioned detectors with low TPR (that are not implemented as alerting logic and left as enrichment) are used in hypotheses. This provides the trade-off: we will not implement detectors with low TPR in alerts and crumble under the multitude of FPs that consume the SOC team's investigation capacity.

Detectors with a low TPR are prone to generating many FPs and thus are intentionally overlooked as part of real-time alerting logic. Including them would overwhelm the SOC team with false alerts, reducing their efficiency. However, we do not completely discard the value of these detectors. Instead, we use them to support hypothesis-driven threat hunting, where analysts periodically review the data to look for stealthy (or less obvious) threats that may not trigger high-confidence alerts. Such detectors therefore contribute to threat detection but in a way that doesn't compromise operational capacity. Combining alert-driven detection with hypothesis-driven threat hunting offers a balanced and comprehensive strategy; relying solely on alerts increases the risk of missing threats (false negatives), while relying only on manual hunting slows down detection.

Every threat detected manually is of course analyzed, and automatic detection and remediation logic is introduced (if technically possible). We've already discussed that threat hunting is the base for threat research from which we create our detectors – and there are at least two use cases that make threat hunting indispensable. The first is completely new threats: despite their scarcity, their impact can be enormous. This is because new attacks are extremely expensive to implement and must be profitable to the attacker. The more complex the attack, the more expensive it is to prepare, which means the investment in attack preparation must be offset by profit for the attacker – which directly impacts the damage to the victim. Hence the simple dependence: the more complex the attack, the higher the potential damage, because the attacker must offset investments in the attack.

The second is less obvious but, in practice, very common – living-off-the-land scenarios. There are many behavior patterns that are indistinguishable from legitimate activity and such triggers cannot be filtered out; they must be continuously monitored to ensure they fall within the scope of legitimate user activity, including, when necessary, contacting the users to clarify the circumstances.

To conclude, it is impossible to overestimate the benefits of retrospective search for a modern SOC. We have only scratched the surface of this important SOC process – but it's enough for preliminary understanding, and the details will become clearer in practice.

About Kaspersky

www.kaspersky.com