sysdig

WHITE PAPER

Securing the Cloud with End-to-End Detection

In today's rapidly evolving technology landscape, where cloud environments, containers, and supply chains are driving digital transformation, ensuring robust threat detection has become more challenging than ever before. The rise of multi-layered infrastructures spanning containers, hosts, cloud platforms, identities, and supply chains necessitates a unified approach to security. It is within this complex security landscape that Falco, an innovative open source project, and Sysdig, the original creators of this project, have emerged as pioneers in delivering end-to-end detection capabilities. Falco builds on 20 years of experience in event capture and forensics, starting with Wireshark, and was built with modern cloud architectures firmly in mind.

As organizations embrace cloud-native architectures and deploy applications across diverse environments, they face a multitude of security challenges. Traditional security measures designed for monolithic architectures struggle to effectively protect the dynamic and distributed nature of modern applications. Threats can exploit vulnerabilities across various layers, making it crucial to adopt a unified approach to security and threat detection.

At the heart of Sysdig Secure lies Falco's unified detection engine. This cutting-edge engine leverages real-time behavioral insights and threat intelligence to continuously monitor the multi-layered infrastructure, identifying potential security incidents. Whether it's anomalous container activities, unauthorized access attempts, supply chain vulnerabilities, or identity-based threats, Sysdig ensures that organizations have a unified and proactive defense against these rapidly evolving threats.

In this paper, we examine the multiple layers at which cloud-based applications and infrastructure are threatened by attacks. We look at how these threat vectors can be detected using Falco, and how Sysdig's platform, built with Falco at its core, brings extra security based on its unique capabilities. Through an examination of the SCARLETEEL exploit, we'll see how these capabilities are employed in the context of a real-life breach.

Table of Contents

The Challenges of Emerging Cloud Threats

The cloud threat landscape has evolved significantly over the years due to various factors, such as increased cloud adoption, changing technologies, and emerging attack vectors. Here are some key aspects of its evolution:

1. **Expanded Attack Surface:** As more organizations migrate their systems and data to the cloud, the attack surface has expanded. Cloud environments offer a vast and complex infrastructure with numerous entry points, including web applications, APIs, and user interfaces. Attackers have more opportunities to exploit vulnerabilities in these entry points to gain unauthorized access or disrupt services.
2. **Sophisticated Attacks:** Cybercriminals have developed more sophisticated attack techniques to target cloud environments. Traditional attack vectors such as malware and denial-of-service (DoS) attacks are still prevalent, but attackers have also adopted advanced tactics like cross-account or cross-cloud attacks, container and serverless-specific vulnerabilities, and supply chain attacks targeting cloud service providers.
3. **Misconfigurations & Human Error:** Misconfigurations continue to be a significant concern in the cloud. Improperly configured cloud resources or services can expose sensitive data or provide unauthorized access. Human error, such as weak passwords, insufficient access controls, or accidental exposure of cloud storage buckets, can lead to data breaches or unauthorized data exposure.
4. **Compliance and Governance Challenges:** Meeting regulatory requirements, such as data protection laws (e.g., GDPR) and industry-specific compliance standards, adds complexity to cloud security management.
5. **Cloud-Native Security Challenges:** Cloud-native technologies, like containers and serverless computing, introduce unique security challenges. Misconfigured containers, insecure container images, or vulnerabilities in serverless functions can be exploited by attackers to compromise cloud environments. Securing these dynamic and ephemeral workloads requires specialized knowledge and security measures.
6. **Inadequate Tooling is Falling Short:** Endpoint detection and response (EDR) tooling is a core component of most security centers. And while effective against traditional threats in workstations, visibility gaps and poor protection begin to rear their head as teams attempt to apply these tools to the cloud. EDR is ineffective in the cloud due to multi-dimensional and ephemeral complexity, sheer volumes of data, and the speed at which it changes.

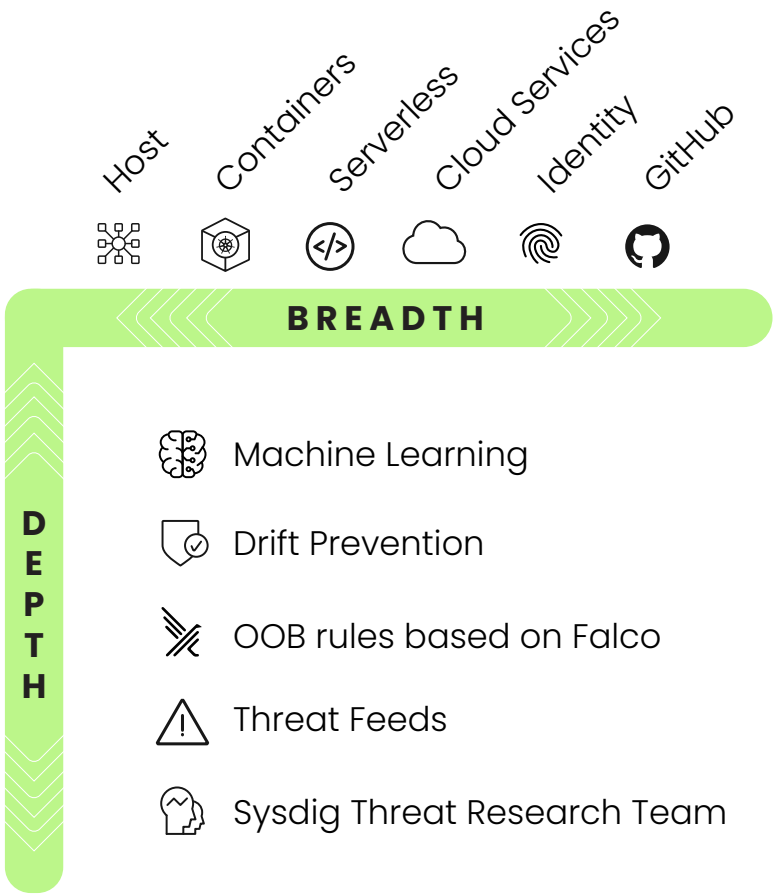
In summary, the cloud threat landscape has evolved significantly over the years, requiring organizations to adopt cloud detection and response (CDR) with a multi-layered approach as part of cloud native application protection platform (CNAPP). By combining multiple security tools and techniques, organizations can reduce the risk of successful attacks and protect their sensitive data and applications in the cloud.

The Spectrum of Cloud Attack Surfaces

Securing the cloud environment poses unique challenges that require breadth of coverage across the software development lifecycle and depth of analysis to protect against known and unknown threats in the cloud.

From a breadth perspective, it's important to have coverage from servers and containers to serverless environments. Equally important is to correlate that information with what's happening at all times with cloud services, identities, and CI/CD providers. In terms of depth, coverage goes beyond traditional rules-based engines. The dedicated Sysdig Threat Research Team continuously contributes and maintains rules that address the latest and emerging threats. This proactive approach ensures that the detection capabilities remain up to date with the evolving threat landscape. In some cases, updated threat feeds are incorporated to provide real-time protection against newly identified risks.

The deep and broad coverage provided by Sysdig ensures that threats originating from different parts of the infrastructure can be quickly detected through a unified approach and that your security program can remain up to date with the evolving threat landscape. We will address all elements of the below diagram throughout this paper.



Servers / Cloud Hosts / VMs

Attackers may attempt to exploit vulnerabilities or weak security controls on servers or cloud hosts to gain unauthorized access or compromise the underlying infrastructure. Multi-layered detection mechanisms, such as host-based runtime security, log analysis, and vulnerability host scanning, can identify and block suspicious activities, detect anomalous behavior, and provide real-time protection against attacks targeting servers and cloud hosts.

According to a study by RedHat, over 60% of Kubernetes security incidents were related to misconfigured or vulnerable hosts. That's why it's critical to have deep runtime detection capabilities at the host level. The Falco engine is designed to detect server-based threats while your services and applications are running. In doing so, Falco detects anomalous, suspicious behavior in real time.

Falco's improved architectural design functions as a streaming engine, processing data upon arrival instead of storing it for later action. This allows for real-time detection capabilities on the server. It then independently evaluates each event, rather than generating alerts based on event sequences, which avoids the incurred overhead associated with real-time detections. Finally, Falco aims to evaluate rules as close to the data source as feasible, minimizing the need for data transport and ensuring alerts can be triggered as soon as possible.

Falco also focuses heavily on data enrichment for host-based detections. One of the most significant examples of data enrichment is when using system calls (syscalls) as a data source. Since syscalls are essential to every application, they occur in just about every context. Data directly provided by a syscall is valueless without context, and therefore it becomes critical to collect and connect surrounding data.

In the case of a mining attack on a server, it's desirable to include Operating System (OS)-specific context in our detections. Information such as processes and threads, file descriptors, users, and groups can all be analyzed through the Falco data enrichment engine. This set of information represents the basic metadata that enables a rule author to make a syscall event useful. Think about it; how would you use a numeric file description in a rule for cryptomining binaries? A filename like 'xmrig' is much better!

How Sysdig helps: Sysdig extends and enhances Falco's capabilities by offering a scalable and comprehensive security platform with a focus on runtime security for server,container, Kubernetes, and cloud logs. Sysdig's Threat Research Team also automatically delivers specially created, optimized, and curated Falco rules, easing operationalization workflows. Investigation and response are enhanced through forensic activity analysis from interactive events and created captures.

Containers & Kubernetes

Containerization has become popular in cloud environments, but it introduces its own security considerations. Attacks targeting containers aim to exploit vulnerabilities in container runtimes, misconfigurations, or insecure container images. Multi-layered detection mechanisms for container security, including image scanning, runtime protection, and behavioral analysis, help identify and block malicious activities, detect container-based attacks, and enforce security policies to mitigate risks associated with workload attacks on containers.

Misconfigurations in the cloud are generally a higher severity issue than we've come to expect in on-premise assets. This is due to the absence of the perimeter. When misconfigurations occur in on-prem assets, they generally affect the specific system or server within the organization's network. While these misconfigurations can still have serious consequences, they are often contained within the network perimeter. On the other hand, cloud environments often provide self-service capabilities for infrastructure such as containerized workloads, allowing for rapid provisioning and configuration of resources. While this agility is beneficial, it can also increase the likelihood of misconfigurations. Without proper controls, mistakes or oversights during configuration can have severe consequences.

While Kubernetes has introduced several concepts to simplify container management and scalability, securing containers using traditional security tools has become more challenging. By now we know that Kubernetes offers a rich set of abstractions, such as pods, services, deployments, and namespaces to streamline workload management. However, gaining comprehensive context usually requires connecting to the Kubernetes Audit logging service.

In the case of Falco deployments, each node in the cluster runs a Falco sensor. During startup, these sensors connect to the API Server to collect cluster data and establish the initial local state. Similar to data enrichment from the Host OS, Falco enhances the log data to provide unparalleled context for the container. Rather than just seeing the process running on a container ID while it was alive, Falco can also tell you which pod, namespace, and deployment it was associated with — regardless of whether the pod is dead or alive.

Also, given the dynamic nature of cloud-native environments and legacy practices carrying over to cloud environments, teams often neglect immutability best practices and are blind to drift, especially at scale. If you're hearing this phrase for the first time, the immutability principle is inherent in cloud-native environments and can often prove advantageous for security teams. Immutability ensures that containers remain unaltered throughout their lifecycle, eliminating the need for updates, patches, or configuration changes.

In the case of environments like Kubernetes, killing a container is different to killing other virtual infrastructure such as Virtual Machines. Due to the ephemeral nature of containerized workloads, if a pod is killed, a brand new container is created in a brand new pod. Immutability ensures the same state is recreated each time. If a change happens inside a running container, it is considered a drift, and possibly a sign of an attack.

Being able to track exactly what drift activity happened inside a container and correlating this with the Kubernetes-layer context from data enrichment is crucial. Only then can we know what kind of security incident occurred, and to which Kubernetes namespace the incident was present in.

How Sysdig helps: Sysdig Secure can prevent, detect, and respond to container threats in real time. Sysdig's Drift Control goes one step further and blocks newly-detected executables from running. This proactive measure helps organizations avoid the risks of malicious code execution at runtime.

Serverless

Serverless computing brings unique security challenges, including potential risks to the underlying infrastructure and code execution environment. Attacks on serverless environments may involve attempts to manipulate serverless functions, execute unauthorized actions, or exploit vulnerabilities in the function's code.

Multi-layered detection mechanisms for serverless security, such as code analysis, behavior monitoring, and anomaly detection, can identify and respond to workload attacks on serverless environments, ensuring the integrity and security of serverless applications.

As cloud platforms have evolved, both the convenience and the abstraction levels have increased simultaneously and new agent models are required.

For example, with Amazon's ECS and EKS, users remain in charge of managing the underlying virtual host machines. In environments like Fargate, however, the hosts are implicitly allocated by the cloud provider and users simply run their containers without allocating, configuring, or having any knowledge of the underlying compute infrastructure.

While this "Container-as-a-Service" model is convenient, it can introduce risk, as many users leave the containers unattended and don't monitor for security events inside them that can exfiltrate secrets, compromise business data, and increase their AWS/cloud provider costs. Furthermore, the additional abstraction layers associated with a serverless architecture essentially hides the underlying infrastructure from the end user.

Unfortunately, for most traditional Endpoint Detection & Response (EDR) tools, without access to the host, visibility into workload activity can be limited in serverless environments. For these reasons, it makes sense for projects like Falco to build their instrumentation to work with AWS Fargate.

In fact, Sysdig's implementation of Falco is the only solution to date that provides a centralized AWS Fargate orchestrator agent required to effectively manage all policy, connections, and events to and from the specific AWS Fargate tasks that provides the deep level of visibility we need to secure AWS Fargate tasks.

How Sysdig helps: Sysdig's serverless workload agent is installed in each Fargate task. It monitors the serverless workload and enforces the Falco policies and rules to detect and prevent security threats, as well as compliance violations.

Cloud Services

Cloud misconfigurations are among the most common security risks in cloud environments. Attackers often target misconfigured access controls, weak authentication mechanisms, and improperly managed storage resources to gain unauthorized access or extract sensitive data. Multi-layered detection helps in identifying and mitigating these misconfigurations by employing various mechanisms, such as configuration monitoring, vulnerability scanning, and policy enforcement.

Regarding how you can go about collecting event data, there are usually two architectural choices:

1. You can query the cloud APIs or watch your cloud data stores to detect misconfigurations.
2. Or, you can stream cloud logs into a backend, index it, and let users query logs.

If the intention is to detect threats in real time, then the first option just isn't good enough. The only real benefit of polling is in cases where you need to perform compliance reports or validation checks. This type of behavior could definitely be performed on fixed intervals, but this certainly ignores the real-time nature of incident response teams who need to detect/stop threats as soon as possible.

The second approach mirrors that of a SIEM solution. A SIEM tends to ingest all possible events, which requires a huge amount of storage for aggregating those events for relevant alerting. While the polling option of querying the relevant cloud API's would certainly require less storage than the SIEM solution, it also lacks the real-time nature of alerting. That's why Falco is the perfect compromise in production systems.

The alternative Falco approach, not listed above, provides an effective way of detecting real-time threats without the cumbersome overhead usually associated with real-time event streaming. Falco parses data in a streaming fashion to detect threats in real time, then it implements detection on top of an engine that is incredibly lightweight to run and deploy. Finally, it offers a compact rule language that is flexible and expressive. Falco consumes few resources and, most importantly, analyzes the data in a streaming way. There is no need to perform expensive copies or wait until the data is indexed.

How Sysdig helps: By incorporating both agentless and agent-based approaches, Sysdig strikes a balance between the ease of initial deployment and speed, as well as the requirement for deep visibility and stronger security as security teams progress in their cloud security journey. For more information on the benefits of agent-based and agentless detentions, check out our **whitepaper dedicated to the topic**.

Identity

Identity-based threats involve unauthorized access, misuse, or compromise of user accounts, credentials, or privileges within the cloud environment. Multi-layered detection mechanisms for identity-based threat detection include user behavior analytics, anomaly detection by analyzing cloud logs, and IAM logs (e.g., Okta). These mechanisms help identify suspicious activities, detect compromised accounts, and enforce access controls to prevent unauthorized access and insider threats.

In March 2022, a cybercriminal group called LAPSUS$ claimed to have successfully hacked into Okta, which is an identity platform used by over 15,000 companies. This security breach occurred only two months before the announcement, leaving Okta's customers unsure about whether their sensitive data had been compromised as well. As a result of this security incident, Okta's security team conducted a thorough investigation and later released some details about the attack.

Okta's identity services are designed to ensure that only authorized individuals can access networks and resources within an organization. This type of service is particularly crucial for companies that deal with sensitive information, such as financial institutions or healthcare organizations. However, this security breach has once again highlighted the importance of detecting suspicious activities as soon as they occur within an organization.

Early detection can help companies mitigate the potential damage caused by a cyberattack, such as data theft, financial loss, or reputational damage. It is essential for companies to invest in robust cybersecurity measures to protect their systems and data against increasingly sophisticated cyber threats.

Traditional cloud security posture management (CSPM) solutions report on scheduled intervals, which means that they may not detect rate limit issues until after the damage has already been done. This approach is not useful when addressing rate limit issues because it does not provide real-time visibility into the authentication system. As a result, security teams may miss critical security events that could lead to a breach.

In the case of Falco plugins like Okta, they offer additional field extraction capabilities for events that were provided by other plugins or core libraries used in Falco. In layman's terms, this allows Falco to implement logic to return the values of data source fields, thus providing additional enriched metadata. Incident responders can now better understand the entire attack, from a compromised user to the impact in cloud-native workloads, by stitching Okta events with real-time cloud, container, and host activity.

How Sysdig helps: Sysdig can prioritize and trim excessive permissions based on analyzing runtime access patterns. Sysdig can also detect identity attacks such as MFA fatigue (spamming) and account takeover (ATO) with the Sysdig Okta detections, and help teams understand the entire attack from user to impact by stitching Okta events with real-time cloud and container activity.

Supply Chain

Supply chain attacks have become a significant concern in recent years, where attackers compromise trusted components or software during the development, distribution, or deployment process. These attacks can introduce malicious code, backdoors, or vulnerabilities into cloud systems. Multi-layered detection mechanisms are essential to identify and mitigate supply chain attacks by analyzing the integrity of software and components, performing code analysis, and leveraging threat intelligence to detect and respond to supply chain compromises.

However, many organizations fail to implement basic security measures to safeguard their source code repositories. For example, they may not restrict access to repositories only to authorized personnel or may not use two-factor authentication to secure accounts. Additionally, some organizations may overlook the risk of sensitive credentials, such as secret keys being pushed to public repositories or even when a private repository is accidentally changed to public. These oversights can leave businesses vulnerable to data breaches and cyber attacks.

One common issue is that businesses may not have a comprehensive security plan in place that includes GitHub repositories. The risks and vulnerabilities of source code repositories can change quickly, and companies must keep up with these changes to avoid becoming a victim of cyber attacks. This is where a dedicated Threat Research team can play a critical role in identifying the latest threats in the wild and updating security rules to detect malicious behavior in GitHub.

Ensuring the security of your team's GitHub repositories is paramount, and one of the most critical aspects of this is detecting and preventing secret exposure in your CI/CD pipeline. The consequences of exposing secrets, such as passwords, tokens, and API keys, can be catastrophic, allowing hackers to gain unauthorized access to your sensitive data.

While leaking secrets in public repositories is an obvious concern, private repositories are just as vulnerable to attacks. Exploiting secrets in private repositories can lead to privilege escalation and lateral movement, posing a significant risk to your team's security. In some cases, as demonstrated **here**, major car manufacturers can make the simple mistake of exposing a "Secret Key" publicly, which can lead to high-profile data breaches.

Unfortunately, detecting and preventing secret leaks can be challenging, particularly for larger teams. It's difficult to control how each member accesses Git and what they commit, and even with tools like AWS's git-secrets, it's essential to install them on every client to be effective. Additionally, even if you remove secrets, they can still appear in your repository's history, making complete deletion impossible.

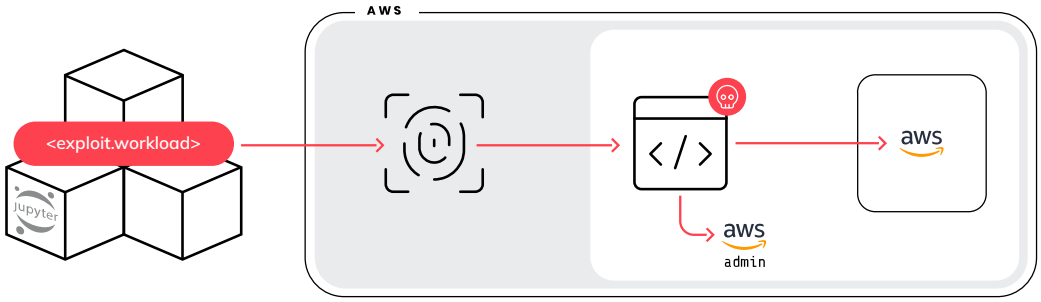
How Sysdig helps: At Sysdig, our researchers identify the latest threats and attack trends, and update the security rules to detect malicious behavior in GitHub as the security landscape evolves. These security rules can help organizations to detect and respond to potential threats in real time, ensuring that their source code is secure and their business is protected from cyberattacks.

Real Life Breach: SCARLETEEL Walkthrough

Attack Step 1: Data Exfiltration via Containers

The Sysdig Threat Research Team discovered a sophisticated cloud operation, dubbed SCARLETEEL. The attacker exploited a containerized workload and then leveraged it to perform privilege escalation into an AWS account in order to steal proprietary software and credentials.

This attack was more sophisticated than most, as it started from a compromised Kubernetes container and spread to the victim's AWS account, proving the real-world need for end-to-end detection with context enrichment from container, Kubernetes, and Cloud. In the SCARLETEEL attack, the attacker found and exploited an internet-exposed web application (service) deployed in a Kubernetes cluster. Once they accessed the container, they started performing different actions to proceed with their attack.



Attack Step 2: Mining Detection in Host and Containers

As you can see from the above diagram, the purpose of the attack went far beyond exploiting a web application to perform cryptomining. Since cryptomining is a well-known threat in virtually all environments (Containers, Hosts, and Serverless), there's a good chance the attackers would use the mining activity as a decoy to evade detection of any high-value activities, such as data exfiltration of proprietary data.

Attack Step 3: Defense Evasion in the Cloud

The SCARLETEEL attackers were able to gain access to sensitive credentials from S3 buckets and Lambda functions, which allowed them to move laterally. We can track all this activity within AWS Cloudtrail logs, but what happens if an attacker tries disabling the cloudtrail audit service in order to evade detection? Thankfully, Sysdig and Falco can detect suspicious user activity in the cloud, such as disabling CloudTrail logging. By enriching data from containers, hosts, Kubernetes, and cloud, we can stitch together a story with context.

Attack Step 4: Authentication Login Failures in Identity Providers

While not explicitly stated in the SCARLETEEL attack, many adversaries bypass additional authentication controls in order to gain access to internal infrastructure. As a result, it's just as important to detect when a user is hitting the rate limits — for example, on OKTA requests. This could indicate a potential threat to the security of the organization's resources. An attacker could use brute force attacks to guess passwords or gain unauthorized access by overwhelming the authentication system with a high volume of requests. From an end-to-end security viewpoint, you also need to detect real-time activity in the tools that secure your cloud.

Attack Step 5: Exposing Sensitive Credentials in the Supply Chain

As seen in the SCARLETEEL attack, Terraform state files contain all data in plain text, which may have secrets. Storing secrets anywhere other than a secure location is never a good idea, and definitely should not be put into source control! The same logic should be applied throughout the shift-left methodology. If sensitive credentials are accidentally committed to a GitHub repository, how exactly do you know this has happened? Do you wait on a colleague to inform you of this insecure commit? And if so, how long after the commit are you made aware of this? Real-time detection is required from the supply chain to container workloads to ensure back actors cannot access sensitive data that was incorrectly exposed in an insecure location.

Conclusion

It is imperative, from a market perspective, to go beyond mere system call detection in hosts and containers. To effectively safeguard against ever-evolving cloud-based threats, it is crucial to enhance the data by incorporating the context of Kubernetes extractions, cloud audit logs, source repositories, and identity providers. The ingestion of logs from multiple sources becomes indispensable when it comes to real-time threat detection and response across the entire stack.

In today's market landscape, complex and sophisticated attacks necessitate comprehensive detection and response capabilities spanning the entire stack. Take incidents like SCARLETEEL, for example, where ingesting logs from various sources alone won't suffice. To achieve high scalability with minimal resource overhead, it becomes imperative to maintain limited state and avoid centralized storage. As exemplified by Falco, the only truly scalable approach to cloud detection and response is to distribute rule evaluation horizontally, ensuring a scalable and effective solution.

Falco's Role In Detecting Cloud Threats

Falco, a recent **graduate project of the CNCF**, is an open source solution for runtime security in hosts, containers, Kubernetes, and cloud. Falco's capabilities serve as the core engine for comprehensive threat protection. The end-to-end detection capabilities encompass real-time detection, multi-layered context, customizability, an active community and ecosystem, and extensibility. These aspects will be discussed in detail below:

1. **Real-Time Detection:** The Falco architecture is unique in that it performs real-time stream processing of data, rather than relying on traditional post-processing methods which incur delays and additional costs. Its eBPF-powered instrumentation provides deep real-time visibility into all system calls across all cloud workloads (e.g., VMs, containers, and serverless) to detect threats, even those in short-lived containers that, on average, live less than five minutes.
2. **Multi-layered Context:** Falco has deep context coming from a variety of sources, such as container runtimes (e.g., Docker), orchestrators (e.g., Kubernetes), cloud provider metadata, and identity providers (e.g., Otka) to enrich the detections.
3. **Customizability:** Falco is highly customizable, allowing users to tailor it to their specific needs and environments. It can be configured to detect and alert on specific behaviors or activities that are relevant to an organization's unique threat landscape. This flexibility enables organizations to adapt and refine their detection capabilities as the threat landscape evolves.
4. **Active Community and Ecosystem:** Falco benefits from the collective knowledge and expertise of a diverse community, leading to a broader set of use cases, integrations (e.g., plugins), and best practices. This active community ensures that Falco remains up-to-date and responsive to emerging threats.
5. **Extensibility:** Falco's plugin architecture allows for extensibility and customization by enabling users to develop and integrate additional functionality into the Falco runtime. It provides a framework for creating plugins that can enhance Falco's capabilities, introduce new rules, and integrate with external systems (cloud, identity, supply chain, third-party app logs, etc.).

Falco offers the best end to end detection approach for a few reasons. It provides breadth across any workload, cloud, or service, allowing organizations to maintain consistent threat detection and visibility regardless of their cloud infrastructure. It also offers depth that comes from combining a variety of techniques, such as machine learning (ML), rules, threat feeds, etc., enabling it to detect and respond to threats effectively.

This combination enhances its accuracy in identifying suspicious activities and provides organizations with a robust defense against both known and emerging threats in their cloud environments. Having created Falco and made it publicly available through contribution to the Cloud Native Computing Foundation, Sysdig has placed it at the core of its comprehensive CNAPP platform.

About Sysdig

In the cloud, every second counts. Attacks move at warp speed, and security teams must protect the business without slowing it down. Sysdig stops cloud attacks in real time, instantly detecting changes in risk with runtime insights and open source Falco. We correlate signals across cloud workloads, identities, and services to uncover hidden attack paths and prioritize real risk. From prevention to defense, Sysdig helps enterprises focus on what matters: innovation.

To learn more about Sysdig, visit sysdig.com

REQUEST DEMO →

WP-008 REV. B 03/24