An Autonomic Approach to Security Incident Response and Prevention in Cloud Computing

Glenn Russell, Roy Sterritt
School of Computing
Ulster University
Belfast Campus,
Northern Ireland
email: russell-g6@ulster.ac.uk |
r.sterritt@ulster.ac.uk

***Abstract*—An autonomic approach for responding to security incidents is proposed, which aims to replace traditionally people intensive, reactive, and technically complex methods for responding to security incidents. In addition, the approach provides the ability for systems to evolve in response to the nature of the attacks, building an immunity iteratively based on real environmental conditions. The solution works alongside existing systems and controls, addressing failures to resolve the complexities of security engineering in heterogenous systems spanning endpoints, traditional data centres, private cloud, and public cloud.**

***Keywords*—autonomic computing; security; infosec; incident response; cloud computing; control.**

I. Introduction

Cloud computing is a paradigm for providing information technology infrastructure and services to users who do not want to own and operate their own physical equipment and want to be able to deploy and scale their applications at will. There are many benefits to this, which have led to ever increasing adoption of cloud computing services, at the expense of more traditional data centres. It was predicted that 2022 [1] would see spending on public cloud services of $482.155 billion, an increase of 21.7% over 2021, and an increase of 53.6% over 2020. There is no sign of this increase in spending abating. IBM [2] lists the benefits as:

- **Flexibility** allows services to be accessed and scaled to fit ever changing demands, from anywhere on the Internet.
- **Efficiency** means that users do not need to spend money on physical equipment, much of which may be redundant, while being able to bring applications to market quicker.
- **Strategic value** is derived from having access to the latest technology as it becomes available, from new processors to the latest machine learning platforms.

Private cloud is where a company or entity make use of their own networking and compute to provide services to users via the use of virtualisation technologies, such as OpenStack [3]. This allows services to be provisioned using an Application Programming Interface (API), then torn down again via the same API when the services are no longer needed. Like private cloud, *public cloud* aims to provide flexible and scalable resources to users, but this service is provided by a third-party, such as Amazon, in the form of Amazon Web Services (AWS) or Google in the form of Google Cloud Platform (GCP).

Services provided by cloud match those available in traditional data centres, but are categorised into several distinct areas. *Infrastructure as a Service (IaaS)* provides basic computing infrastructure in the form of virtual machines, networking, and storage. This is a core element of all clouds, both public and private, and has arguably [4] become increasingly commoditised. AWS have the Elastic Compute Cloud (EC2) service for virtual machines, while GCP has Google Compute Engine (GCE) providing the same service, as just two examples. *Platform as a Service (PaaS)* is an abstraction which prevents the user from needing to manage compute directly, and instead provides a framework for building and deploying applications. As part of this, advanced features, such as identity, access management and security are usually provided. Examples of this are Google App Engine, Heroku and Vertex AI [5]. *Software as a Service (SaaS)* removes practically all responsibility from the user of managing and running an application (other than managing user access), and instead provides direct business value. Examples of this are Salesforce, public GitHub, and Google Docs [6]. Finally, *Container as a Service (CaaS)* refers to a service which orchestrates many different sub-components running in containers into a fully managed distributed application. While the term is agnostic of any given implementation of the technology, this usually refers to Kubernetes [7], a platform developed by Google to manage massive applications. Each of the major public clouds offer managed Kubernetes services; GCP provides Google Kubernetes Engine (GKE), Azure provides Azure Kubernetes Service (AKS), and AWS provides Elastic Kubernetes Service (EKS). Out of the various service offerings offered in cloud, CaaS is the closest to offering autonomic capabilities. Casalicchio argues [8] that container orchestration does not include any autonomic features because of a reliance on hypervisors and simplistic heuristics for actions like scaling, but this misses two key points. The hypervisor is not the autonomic management agent in a CaaS, it is the master node [9] along with the *Kube-Controller-Manager*, and secondly that the process of scaling pods and nodes is already autonomic in the case of managed offerings, such as GKE.

Underlining this shift in how Information Technology (IT) services are deployed and managed are the kind of workloads being run on clouds. They are no longer the reserve of small, rapidly innovating start-ups, but are used by over 90% of the largest companies in the world [10]. Large financial institutions like CapitalOne closed the last of their data centres in 2020, relying entirely on Amazon Web Services to run their entire I.T. estate [11]. However, with this seismic shift in how services are run, so have these new services been exposed to new kinds of threats. While a threat is often thought of as a malicious actor, whether that be a script kiddie or hacker collective, the most significant cause of breaches is human error. In fact, IBM found [12] that human error is the root cause in 95% of cases. The combination of simple to deploy services with complex and difficult to fully understand API configurations means that even before services are deployed, vulnerabilities are already built into a service. With the ease with which new Tools, Techniques, and Procedures (TTPs) are brought to bear by attackers on the Internet, security practitioners are facing multiple threats from internal and external vectors. This has led to both a shortage of trained cyber professionals [13] and burnout among existing people [14]. Clearly, the burden on cyber professionals is increasing, with 2021 seeing a 1885% increase in ransomware attacks alone [15].

Security in IT systems (often referred to simply as *Cyber*) is a central concern in how people conduct their lives, how nations and governments run their countries and manage their societies. The digitisation of society, while providing an unprecedented level of access to information and communication, has introduced an equal and opposite issue in exposing our society to threats that transcend both physical, national, and geographical boundaries.

Hagen et al. [16] refer to the challenges of physical distance, borders and time diminishing. However, the impact of this is that society has become much more susceptible to various kinds of malignant activity from threat actors that ranges from causing reputational damage in the form of web site defacement, hacktivism from organisations like Anonymous [17], or even attack from nation states. Cryptographer and security expert Bruce Schneier predicted [18] the rise of rapid automated attacks, perpetrated from a distance, and the subsequent proliferation of these techniques would require only a single skilled threat actor, while the ability to communicate at will over long distances in secret would mean that a technique could simply be copied by others. The prevalence and effectiveness of cyber-attacks by nation states has resulted in a move to defensive postures that will possibly include what is euphemistically called a kinetic response – a cyber-attack could soon lead to a physical military response [19]. Given the state of current computing paradigms and the associated financial and societal risks, methods must be developed which can remove the burden of securing of those paradigms as much as possible from the human.

In Section 2, security in cloud computing is examined, focusing on defense in depth. Section 3 focuses on Autonomic Cloud Computing for security. Section 4 goes into more detail on applying these autonomic principles for security. Section 5 proposes an Autonomic Incident Response System, and finally the paper concludes with Section 6.

II. Security in Cloud Computing

Regardless of the computing paradigm being deployed, effective security programmes adopt some fundamental principles, which can be utilised regardless of the computing paradigm. The principle of these approaches is *defense in depth* (Figure 1), which calls for a series of defensive mechanisms which are layered to protect valuable data and information. Having multiple layers of security ensures that there are redundant controls in place if a specific control is compromised [20]. However, these layers are typically not environment aware. An example here is a firewall that is used to block non-authorised network traffic, which protects a virtual machine, which runs anti-virus software, which is detecting malware. Neither security control is aware of the other, or shares information about their environment. This is a major shortcoming and prevents contextual knowledge from being shared to protect other assets if a machine is infected with malware. Ideally, the anti-virus agent would let the firewall know that a piece of malware would try to attack other assets (this process is referred to as *traversal*) by passing metadata, which notes a particular traffic of network traffic using five-tuple [21] data along with a file hash as part of a message payload. The firewall could then react to the attack in real-time. At the same time, the malware metadata could be used by another component, such as a malware sandbox where it could be detonated to provide further data to further enhance defensive measures, or even to maintain a chain of custody for forensic analysis of a breach. This process is part of what is referred to as *incident response*, which is a procedure for dealing with a security incident.

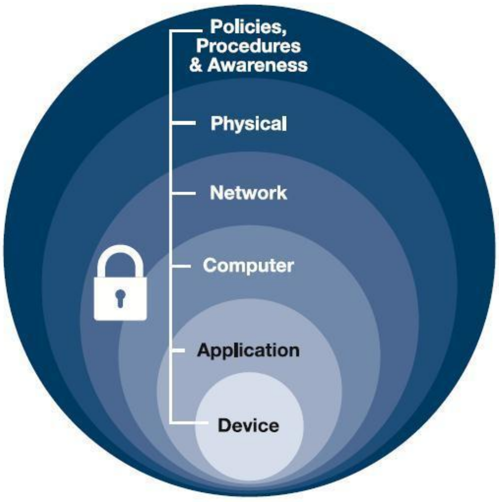

Figure 1. Defense in depth.

The scenario described is already understood by security practitioners (though this is a simpler use case), but there are two issues which make this process very resource intensive and increasingly unmanageable:

- The incident response process is manual, meaning it is very resource intensive and requires specific expertise.
- There are many of these incidents per day, and too few people to respond. This has led to the prevalence of a condition called *alert fatigue*, which is the scenario where security teams have too many alerts to be able to work effectively [22].

The security industry has attempted to resolve these core issues by introducing new kinds of tools and automation to make the job of the security team easier. There are several classes of tools to achieve this. An ad-hoc nomenclature exists which describes the types of security data that are used in cloud and security platforms. An *Indicator of Compromise* (IoC) is a digital artifact, such as a file, hash or configuration that is a sign of an attack. These can be shared among systems so that they can all be protected by detecting an attack. Organisations called *Information Sharing and Analysis Centers (ISACs)* exist for industry verticals where trusted partners can share these using threat sharing platforms, e.g., the *Financial Services ISAC (FS-ISAC)*. A *vulnerability* is a package or system misconfiguration that is susceptible to attack, and lastly an *exploit* is a piece of software or technique that can be used to take advantage of a vulnerability. *Security Information and Event Management (SIEM)* platforms perform two main functions. Firstly, they collect log and event information from networks and devices and store the data so that it can be searched. Logs could take the form of access logs, network traffic information, web requests or DNS requests. All this log data enables the second purpose of these platforms, which is to look for anomalies in the logs that may be indicative of a breach or attempted attack, both in real-time and as part of a forensic analysis of a breach. For example, if a log message indicates a particular IP address is performing thousands of requests a second, it may be indicative of a DDoS attack. Rules are written which describe these conditions, and when these conditions are detected, alerts are raised which are handled by a human operator. It does so in real-time, but as the volumes of logs have increased exponentially, these platforms are having trouble scaling to meet demand. *Threat intelligence* seeks to augment a SIEM by providing information about malicious sources, which can be then used in real-time to filter alerts, reducing the cognitive load on the security practitioner. Practically, if a SIEM has a known list of IP addresses that it knows are a source of malicious traffic, then it can prioritise alerts on those rather than attempting to filter and analyse all sources.

Integrating SIEM, threat intelligence and other tools, such as Endpoint Detection and Response (EDR) tools together and providing procedures for responding to threats are *Security Orchestration, Automation and Response (SOAR)* platforms Repetitive processes can be handled automatically, and a SOAR platform could be used to respond to a malware attack as described, or to automatically shut down a virtual machine if it is found to be infected with a critical vulnerability. Like a SIEM, these rules or playbooks must be manually written for the platform to be effective. With so many different tools and techniques, there is a significant challenge in simply being able to integrate them. The single common standard for sharing vulnerability information for many years has been the *Common Vulnerabilities and Exposures (CVE)* standard [23], which is how software vulnerabilities, such as Heartbleed [24] are communicated for consumption by human and machine alike. It describes details, such as whether the vulnerability can be exploited over a network or without authentication to the host system. In recent years, several other standards have emerged under the stewardship of Oasis in the form of the *Structured Threat Information eXpression (STIX)* standard, and complementary *Trusted Automated eXchange of Indicator Information (TAXII)* standard. These formats are XML-based and are used to describe all manner of threats, such as malware or network-based attacks, independent of any single vendor or implementation, and indicators of compromise using the embedded CyBOX standard. The *Security Content Automation Protocol (SCAP)* is a standard proposed by NIST [25] that allows for automated vulnerability management and is in use by many major security solutions. *Common Vulnerability Scoring System (CVSS)* is an important standard because it attempts to add a dynamic weight to a vulnerability through its *environmental* score. If a vulnerability is exposed directly to the Internet, then the weight is increased to reflect the higher risk of exploitation, or if mitigated by a network control, it is greatly reduced for the opposite effect. Other standards exist expressing similar data, but in summary, there are many standards that support integration of various security tools and processes.

The integration of various log sources, security platforms and controls together are an ideal outcome, which in theory should produce an effective immune system that can detect and respond to threats more effectively. The reality is far from the truth. The result of efforts by security vendors to solve these many problems has resulted in an explosion in complexity of security tools which require all new skills to be able to operate and interpret. In fact, deploying new security tools may not improve security at all, but have the opposite effect due to a decreasing ability to detect an attack [26]. Even in the case where a tool adopts an open standard, such as SCAP, if other tools in the environment do not at least support it also, then the ability to integrate is greatly reduced.

III. AUTONOMIC CLOUD COMPUTING AND SECURITY

While autonomic computing is a well-defined domain, it does not hold exclusivity over the main features of an autonomic system, and cloud computing platforms exhibit several features which classify aspects as autonomic. Indeed, Cloud Computing was Autonomic Computing's major impact success during its 2nd decade [27]. These principles are refined into just four, so-called *self-CHOP*.

- Self-*Configuration*
- Self-*Healing*
- Self-*Optimisation*
- Self-*Protection*

Self-configuration is supported by both IaaS and PaaS services to build distributed applications. Etchevers et al. [28] outline various methods to achieve this. Virtualisation and corresponding formats are a focus of the paper which concludes that the formalisms and mechanisms offered by industry are basic, non-exhaustive and non-extensible. A key point from the paper is that vendors are moving towards common APIs, such as OVF to describe applications. However, the paper does not explore the practice of *configuration management*, which addresses the shortcomings identified by providing the ability to autonomously configure multiple applications and multiple fine-grained applications. Popular tools in this space are *Puppet*, *Chef*, *Ansible* and *Terraform*. Each of these tools possess a management component or master which configures new and existing components, such as new servers coming online via an agent or surrogate agent process. Both the master and agents are akin to an *autonomic manager*, which exchange information on the desired state of an environment and the actual state. *Configuration drift* (Figure 2) is where the configuration of a service differs from the expected configuration, and it is this that the master attempts to correct for each service. It does so iteratively through a process called *eventual consistency*, in which the master issues commands over a secure channel to make corrections, and the services (in fact, an autonomic manager) respond with a snapshot of their current state. This continues until there is no configuration drift. This mechanism results in what is a *self-healing* process that can operate with any aspect of the cloud that can be managed programmatically. Configuration management tools can integrate with any aspect of a cloud environment, including security controls. This results in a *declarative* capability to define the desired state of an environment. While this implies a static system, these tools also allow rules to be added which dictate how a system should behave under load or when failures occur, which gives the environment the ability to define *fault tolerance* declaratively and have the managers enact it. Cloud is also *self-optimising* because it can scale many aspects of the environment according to pre-defined conditions and rules. This ability to pre-programme the addition and removal of services from the environment results in *apoptotic* services. Servers may be started (using the autonomic processes available from configuration management tools) based on some collection of metrics, such as requests per second to an application or increased CPU usage. Then when loads fall back under some threshold (which could be static or learned over time) servers are shut down again. The ability to shutdown services in response to some environmental event is a key part of an autonomic security capability. For example, if a server is compromised, it should be quarantined, snapshotted for forensics, then shutdown before it can be used by an attacker to traverse the network. What is lacking in cloud is the integration and proliferation of these events in a way which is standardised and in real-time. The environmental event could come from a SIEM platform or threat intelligence platform using STIX.

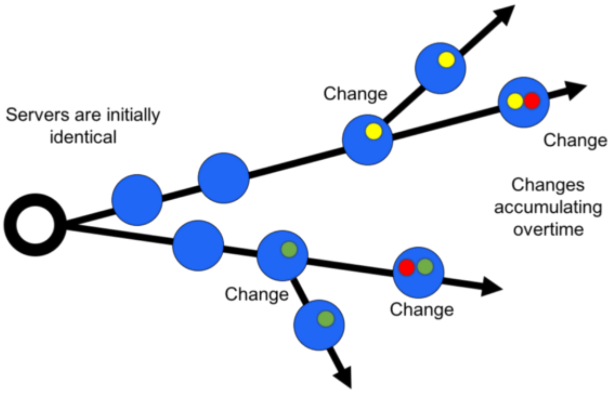
Servers are initially identical
Change
Change
Changes accumulating overtime
Change
Change

Figure 2. Configuration drift.

The ability to declare what a cloud environment should look like and have configuration management processes configure and heal services gives us the ability to build *self-protecting* systems. However, this is where the current state of cloud largely fails to embrace autonomic principles. To self-protect, a component must be aware of internal and external threats, and this complexity is why securing any IT system, not just cloud is becoming exponentially more difficult. Consider the following incident example:

- Configuration manager defines a network and application which runs on port 443.
- The application uses version 10.1 of a web server.
- A new server starts up and it's agent communicates with the master to retrieve it's configuration and install the application running on port 443.
- The new server reports back to the configuration manager which compares declared versus actual state, sees they are the same, so no further action is taken.
- Thirty minutes later, the new server checks again with the master to compare declared state. They are the same, so no changes are required.
- A new zero day exploit on the web server being used is found. There is no patch available yet, but metadata is available.
- An application is running for which there is no defence yet.

There are several possible mitigations for this. The use of a SOAR platform may be able to automatically shut this service down or create a firewall rule that blocks traffic to this port. The issue is that the SOAR does not know about the zero day to be able to take an action in the first place. A threat intelligence feed could provide this information automatically, but there will always be a lag between a zero day being found and exploited and the time it is detected and mitigated. Even the associated CVE may not contain enough metadata to be assist an autonomous security platform, and there is often a lag of days or weeks before information is available in the National Vulnerability Database (NVD), examined in detail by Ruohonen [29]. In addition, in real-world scenarios, services are simply not shut down and security controls automatically configured without oversight, due to the risk of business outages or even inadvertently introducing even more vulnerabilities into a system. Security platforms, such as IBM QRadar Risk Manager [30] disabled features which allowed the automatic configuration of security controls for these reasons. Even in an environment where a mature configuration management strategy is in place, security tools are in place and well-tuned, incidents cannot be responded to in real-time.

IV. Applying Autonomic Principles to Security

Clearly, there are significant efforts to bring the power of automation to bear on the dual problem of ever-increasing complexity, and ever more scarce resources when it comes to dealing with it. Cloud has nascent support for an autonomous approach to security in the form of configuration management tools APIs (though unique to each cloud implementation). Defence in depth strategies call for multiple independent components working together to provide layered security, and while individual controls are effective at addressing specific kinds of controls, such as protecting a web application or defending a network, they are not *context-aware* because they do not understand the environment in which they are operating. (This means that if a vulnerability is present, it is not clear how critical it is. It could be hidden behind many other controls or exposed directly to the Internet). This is a key requirement of eight conditions that IBM define [31] as features of an autonomic system, which are:

- The system is aware of the resources it can access and why it is connected to other systems.
- It can automatically configure itself based on its environment.
- It must be able to optimise itself for efficiency.
- It must be resilient in the face of problems through self-healing or avoiding issues.
- It must protect itself against attacks.
- It must adapt to its environment by establishing connections with adjacent systems.
- It should rely on open standards.
- It can predict demand for its resources and adapt in a manner transparent to other systems.

Research in autonomic security is relatively non-existent but is gaining momentum in industry. Google have labelled their own initiatives as the "10x SOC", referring to a *security operations center* which can be considered the central nervous system in an enterprise. The focus of this effort is to address the issues set out by this paper, in terms of throughput achieved over current methods [32]. As is predictable for a vendor publication, prominence is given to specific products, but nevertheless, it identifies the following building blocks of an autonomic SOC:

Products, including Chronicle, Looker and BigQuery, which mirror the functionality of a SIEM in providing analysis of logs and events.

Integrations to EDR, SOAR, etc.

Blueprints, including network forensics and telemetry.

Content, which includes rules, logs and security detection playbooks.

Despite originating from a deeply technical company like Google, their full paper [33] does not propose an autonomic solution, and falls short of any kind of technical insight into an approach, but it does serve to underline the finding of this paper so far, and that is the components are available to build an autonomous system. What is lacking is cohesion in the form of an *autonomic communications channel* and standardised message formats. This paper has identified several open message formats that can be used to communicate security information between all kinds of components.

Thus, reviewing the eight attributes of an autonomic system and combining it with what has been identified so far in terms of cloud and security technology, we can map the eight autonomic principles as defined by IBM to an autonomic security solution for cloud environments.

- It can be *aware* of systems that it is connected to via the use of configuration management declared state.
- It can *configure* itself and other components via configuration management agents.
- It can *optimise* itself through metrics gathered from the environment via manager components and data generated by the cloud platform to block threats not previously seen.
- It can *self-heal* by turning off infected or compromised hosts using SOAR or restarting services that unexpectedly crash or fail.
- It can *protect* itself by declaring known state, and fixing configuration issues if configuration drift is detected.
- It can *partially adapt* to its environment by using the declared state to understand adjacent systems and use environment information to modify its own behaviour.
- Many *open standards* exist which allow components to communicate, such as STIX, TAXII, SCAP and ATT&CK.
- It can *predict* demand and future events by using environment information and threat intelligence data.

To achieve an autonomic solution, cloud and security technology must operate as a single immune system, rather than as vestigial appendages to one another.

V. AN AUTONOMIC INCIDENT RESPONSE SYSTEM

This paper has summarised the many challenges facing security practitioners as they secure and defend their platforms against both internal and external actors. It has also researched the current state of cloud as a means for managing complex distributed applications. In doing so, a complex and heterogenous landscape of point solutions and loose integrations has been identified which increases complexity rather than reduces it. Proposing a solution which introduces yet another security tool to actively manage will not resolve the issues in a meaningful way. Therefore, the following must be true of any solution:

- The solution must augment existing tools and platforms. i.e., the solution should utilise existing security services or agents as their managed component.
- The solution should adopt autonomic principles in a manner which does not increase the cognitive load on security practitioners. It will do this by automating incident response and cutting the human out of the loop.
- The solution must adopt open standards to enable messages and knowledge to be shared among components of the solution.

An autonomic solution requires that the following components be present in the solution:

- An **autonomic element** which is a combination of a managed component and an autonomic manager
- The **managed component** which in this case could be any kind of security apparatus that we would to managed autonomically, e.g., a firewall or user access list.
- The **autonomic manager**, which operates the managed component based on feedback, such as messages received from the environment.
- Communication between the autonomic elements will be achieved with an **autonomic communications channel**. As part of this, messages will be formatted according to open standards, such as STIX, CyBOX and SCAP.

The **environment** should be considered as the full extent of a cloud deployment which hosts infrastructure that provides some value, of any combination of services. In the case of a website, this could be virtual machines, databases, message queues and an in-memory cache, for example. A defence in-depth strategy (Figure 3) calls for multiple layers of security. The solution proposes that each logical layer of such a strategy is secured by an autonomic element as described.

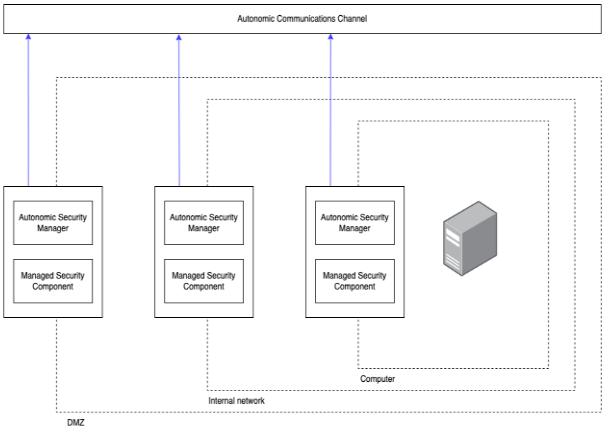
Autonomic Communications Channel
Autonomic Security Manager
Managed Security Component
Autonomic Security Manager
Managed Security Component
Autonomic Security Manager
Managed Security Component
Computer
Internal network
DMZ

Figure 3. Autonomic defence in depth.

Autonomic elements must consume and emit the following kinds of messages:

- **Indicator of compromise** data will be passed between autonomic elements using the OpenIOC standard. This is our *reflex signal*, to which the system is expected to respond, which should result in a mitigation. Examples of this message may be an IoC for a piece of malware identified by an EDR solution (the managed component) and published to the autonomic communications channel by an attached autonomic manager.
- **Vulnerabilities** will be expressed using the CVE format and associated *Common Platform Enumeration (CPE)* format which allows specific operating system, package, and version information to be expressed. An example message would specify that OpenSSL version 1.1.3 on Linux has a critical vulnerability.
- **Mitigations** required for managed components will be passed using the *SCAP* standard, which contains machine readable data expressing how the

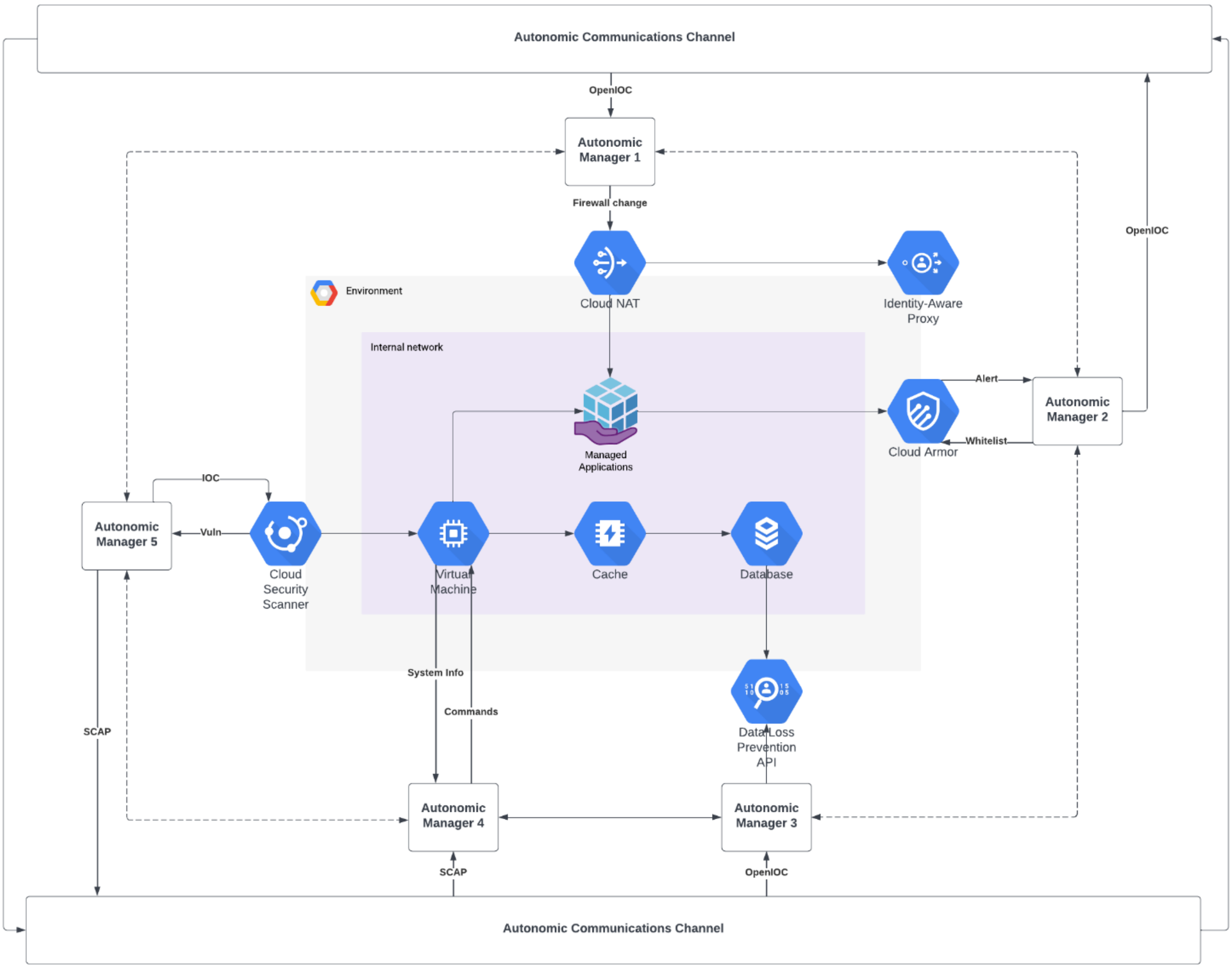

Figure 4. Autonomous secure cloud environment.

environment state should be modified to remediate a vulnerability, possibly an apoptotic response to the reflex signal. An example SCAP message would specify that a certain Windows 11 Pro service should be disabled.

A scalable and fault tolerant message bus, such as Kafka or RabbitMQ will constitute the **autonomic communications channel** which each autonomic manager will both subscribe and publish to. These message queues are built to ensure that messages are always delivered and can scale up to many millions of messages per second, so important security events are guaranteed to be delivered. Each managed component is an existing security control or cloud service. The autonomic manager integrates with it via existing APIs and acts as a gateway between control specific messages and the standardised formats the solution is relying on. While each autonomic element receives every reflex signal being triggered, it is up to each specific element to decide how to react to it, and if it also needs to transmit a reflex signal in turn. By combining both cloud services and security controls into a single autonomous system, an immune system is created which removes the need for a human in the loop because existing security tools integrate poorly with the environment they are protecting.

To understand how the solution would work, consider Figure 4. AM5, which manages a vulnerability scanner, detects a vulnerability on a VM and emits an SCAP message with remedial details. AM4 receives the message and issues a system command which updates the environment state with the remedial action and the change is made as configuration drift has occurred between the desired state and actual state. AM1, AM2 and AM3 receives the message but does not perform any action.

In another scenario, AM3 detects that data is being sent to an unauthorised IP outside of the environment. It emits an OpenIOC message. AM1 receives the message and instantly enacts a change to the environment to block this network traffic. In addition, AM2 receives the message and queries the Cloud Armor Web Application Firewall (WAF) for all traffic sent from the offending external IP address and emits an OpenIOC message. Upon receiving the message AM5 conducts a vulnerability scan of the web applications being hosted that interacted with the external IP, based on the messages from AM2.

At no point in these interactions is a human necessary to perform any action. This fact is the advantage of an autonomic security solution, as the workload on security practitioners is greatly reduced.

VI. Conclusions

So far, the security industry has failed to take advantage of the many features covered by this paper, and only increased the complexity of systems overall, failing to take advantage of autonomic principles in favour of artificial complexity.

Taking a devil's advocate position, a serious ethical issue with the solution is the consumption of data which has historically been heavily biased against network traffic originating in certain regions, and weight that traffic is much more negatively based on this fact alone. In this solution, this will manifest itself in the number of IoCs being flagged as originating in regions, such as Russia or China. The root cause of this is the bias in threat intelligence which is either directly or indirectly consumed by security tools and cloud platforms. This will directly lead to users from those areas being treated differently than others based on an explicit bias. However, the move to purely autonomous security platforms can greatly reduce this issue by removing very real cognitive bias introduced by human operators. Of course, the irony of the tendency to instantly associate any activity from Russian and Chinese sources is that although both these countries are undeniably involved in cyber warfare as nation states [34], in the case of Russia at least they have not launched a mass surveillance and illegal wiretapping campaign to match the scope of that perpetrated by GCHQ and the NSA in the form of the PRISM programme [35]. So, in this case, a very real bias results in blind spots as teams may not consider 'friendly' nation states as potential sources of attack. This is underlined by the simple fact that attempting to search for material associated with nation state threat actors will yield results that are almost exclusively non-western countries. Finally, to underline the effect of cognitive bias, consider the conflict between Russia and Ukraine and its impact on the security practitioner. Given the clear distinction in the roles of aggressor and victim as portrayed in western media, this could result in a human unconsciously giving more weight to a Russian IoC than a Ukrainian IoC. The ability for an autonomous system to operate purely on observations and data effectively negates this very real shortcoming in 'human-in-the-loop' security platforms. Lastly, it is worth noting (at least as of 2013) that Russia was only fourth in the rankings for sources of cyber-attacks, while the US was second [36]. Autonomicity provides the opportunity to remove bias from the system along with its stated aim of intelligent self-management.

ACKNOWLEDGEMENTS

The first author managed development of IBM QRadar Risk Manager [30] product for IBM. Product names may be a registered trademarked by their respective owners, all rights reserved. This paper was produced as part of COM760 Autonomic Computing & Robotics on Ulster University's MSc in Artificial Intelligence.

REFERENCES

[1] Gartner, "Gartner Says Four Trends Are Shaping the Future of Public Cloud," Gartner, 2 August 2021. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-08-02-gartner-says-four-trends-are-shaping-the-future-of-public-cloud. [Accessed 10, 2023].

[2] IBM, "Benefits of Cloud Computing," 10 October 2018. [Online]. Available: https://www.ibm.com/uk-en/cloud/learn/benefits-of-cloud-computing. [Accessed 05, 2022].

[3] OpenStack, "The Most Widely Deployed Open Source Cloud Software in the World," OpenStack, 1 March 2022. [Online]. Available: https://www.openstack.org/. [Accessed 10, 2023].

[4] A. McLean, "Has IaaS commoditisation triumphed?," 2014. [Online]. Available: https://www.comparethecloud.net/editor-recommends/has-iaas-commoditisation-triumphed-over-iaas-differentiation/. [Accessed 10, 2023].

[5] Google, "Overview of Generative AI on Vertex AI," 11 2023. [Online]. Available: https://cloud.google.com/vertex-ai/docs/generative-ai/learn/overview. [Accessed 11, 2023].

[6] S. Dawson, "What Is SaaS? (With 23 Successful SaaS Examples)," 11 2022. [Online]. Available: https://dawsonsimon.com/saas-examples. [Accessed 11, 2023].

[7] CNCF, "Production-Grade Container Orchestration," CNCF, 1 March 2022. [Online]. Available: https://kubernetes.io/. [Accessed 10, 2023].

[8] E. Casalicchio, "Autonomic Orchestration of Containers: Problem Definition and Research Challenges," in *VALUETOOLS'16: Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools on 10th EAI International Conference on Performance Evaluation Methodologies and Tools*, 2016.

[9] R. Mohamed, "Kubernetes Cluster vs Master Node," Suse, 16 April 2019. [Online]. Available: https://www.suse.com/c/kubernetes-cluster-vs-master-node/#:~:text=What%20is%20Master%20Node%20in,the%20frontend%20to%20the%20cluster.. [Accessed 10, 2023].

[10] T. Luxner, "Cloud computing trends and statistics: Flexera 2023 State of the Cloud Report," Flexera, 5 April 2023. [Online]. Available: https://www.flexera.com/blog/cloud/cloud-computing-trends-flexera-2023-state-of-the-cloud-report/. [Accessed 10, 2023].

[11] S. Fregoni, "Capital One closes all data centers, relies on AWS on-demand infrastructure," Silicon Angle, 1 December 2020. [Online]. Available: https://siliconangle.com/2020/12/01/capital-one-closes-all-data-centers-to-rely-on-aws-on-demand-infrastructure-reinvent/. [Accessed 10, 2023].

[12] M. Ahola, "The Role of Human Error in Successful Cyber Security Breaches," usesecure, April 2019. [Online]. Available: https://blog.usecure.io/the-role-of-human-error-in-successful-cyber-security-breaches#:~:text=According%20to%20a%20study%20by,have%20taken%20place%20at%20all!. [Accessed 10, 2023].

[13] J. Legg, "Confronting The Shortage Of Cybersecurity Professionals," Forbes, 21 October 2021. [Online]. Available: https://www.forbes.com/sites/forbesbusinesscouncil/2021/10/21/confronting-the-shortage-of-cybersecurity-professionals/?sh=d27c8f178b9b. [Accessed 10, 2023].

[14] J. Coker, "Stress and Burnout Affecting Majority of Cybersecurity Professionals," Info security group, 2021 September 2021. [Online]. Available: https://www.infosecurity-magazine.com/news/stress-burnout-cybersecurity/. [Accessed 10, 2023].

[15] A. Taylor, "There's a huge surge in hackers holding data for ransom, and experts want everyone to take these steps," Fortune, 22 February 2022. [Online]. Available: https://fortune.com/2022/02/17/ransomware-attacks-surge-2021-report/#:~:text=Governments%20worldwide%20saw%20a%201%2C885,SonicWall%2C%20an%20internet%20cybersecurity%20company. [Accessed 10, 2022].

[16] D. J. Hagen and D. O. Lysne, "Protecting the Digitized Society—the Challenge of Balancing Surveillance and Privacy.," JSTOR, 2016. [Online]. Available: https://www.jstor.org/stable/26267300?seq=1#metadata_info_tab_contents. [Accessed 10, 2023].

[17] V. Karagiannopoulos, "A decade since 'the year of the hacktivist', online protests look set to return," 29 June 2021. [Online]. Available: https://theconversation.com/a-decade-since-the-year-of-the-hacktivist-online-protests-look-set-to-return-163329. [Accessed 10, 2023].

[18] B. Schneier, Secrets and Lies: Digital Security in a Networked World, with New Information about Post-9/11 Security, 2nd edition, Indianapolis: Wiley Publishing, 2004.

[19] K. Townsend, "UK Warns That Aggressive Cyberattack Could Trigger Kinetic Response," 15 May 2018. [Online]. Available: https://www.securityweek.com/uk-warns-aggressive-cyberattack-could-trigger-kinetic-response. [Accessed 10, 2023].

[20] Forcepoint, "What is Defense in Depth?," Forcepoint, 2022. [Online]. Available: https://www.forcepoint.com/cyber-edu/defense-depth. [Accessed 10, 2023].

[21] M. Rouse, "What Does 5-Tuple Mean?," 21st May 2014. [Online]. Available: https://www.techopedia.com/definition/28190/5-tuple#:~:text=Explains%205%2DTuple-,What%20Does%205%2DTuple%20Mean%3F,and%20the%20protocol%20in%20use.. [Accessed 10, 2023].

[22] D. Raywood, "Alert Fatigue and Overload an Issue for Majority of Security Analysts," 9th July 2020. [Online]. Available: https://www.infosecurity-magazine.com/news/alert-fatigue-overload-issue/. [Accessed 10, 2023].

[23] Mitre, "CVE® Program Mission," 2022. [Online]. Available: https://www.cve.org/. [Accessed 10, 2023].

[24] Mitre, "Heartbleed CVE," 2014. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2014-0160. [Accessed 10, 2023].

[25] NIST, "Security Content Automation Protocol," 2022. [Online]. Available: https://csrc.nist.gov/projects/security-content-automation-protocol. [Accessed 10, 2023].

[26] Ponemon institute, "The 2020 Cyber Resilient Organization Study by the Ponemon Institute," IBM, 2022. [Online]. Available: https://www.ibm.com/account/reg/us-en/signup?formid=urx-45839. [Accessed 10, 2023].

[27] R. Sterritt, "Keynote: 20 Years of Autonomic Computing," in *17th International Conference on Autonomic and Autonomous Systems (ICAS)*, Online (Covid-19), 2021.

[28] X. Etchevers, T. Coupaye, F. Boyer and N. De Palma, "Self-configuration of distributed applications in the cloud," July 2011. [Online]. Available: https://www.computer.org/csdl/proceedings-article/cloud/2011/4460a668/12OmNznCl2S. [Accessed 10, 2023].

[29] J. Ruohonen, "A look at the time delays in CVSS vulnerability scoring," 2 December 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210832717302995#b0080. [Accessed 1, 2023].

[30] IBM, "IBM QRadar Risk Manager," 24 January 2022. [Online]. Available: https://www.ibm.com/docs/en/qsip/7.3.2?topic=manager-qradar-risk. [Accessed 05, 2022].

[31] IBM, "An architectural blueprint for autonomic computing," June 2005. [Online]. Available: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf.

[32] I. Ghanizada and A. Chuvakin, "Modernizing SOC ... Introducing Autonomic Security Operations," 21 July 2021. [Online]. Available: https://cloud.google.com/blog/products/identity-security/modernizing-soc-introducing-autonomic-security-operations. [Accessed 10, 2023].

[33] I. Ghanizada and A. Chuvakin, "Autonomic Security Operations," 2 July 2021. [Online]. Available: https://services.google.com/fh/files/misc/googlecloud_autonomicsecurityoperations_soc10x.pdf. [Accessed 10, 2023].

[34] Mandiant, "Advanced Persistent Threat Groups," [Online]. Available: https://www.mandiant.com/resources/apt-groups. [Accessed 10, 2023].

[35] The Guardian, "UK gathering secret intelligence via covert NSA operation," 7th June 2013. [Online]. Available: https://www.theguardian.com/technology/2013/jun/07/ uk-gathering-secret-intelligence-nsa-prism. [Accessed 05, 2022].

[36] N. Knell, "Top 10 Countries Where Cyber Attacks Originate," 23 April 2013. [Online]. Available: https://www.govtech.com/security/hacking-top-ten.html. [Accessed 10, 2023].