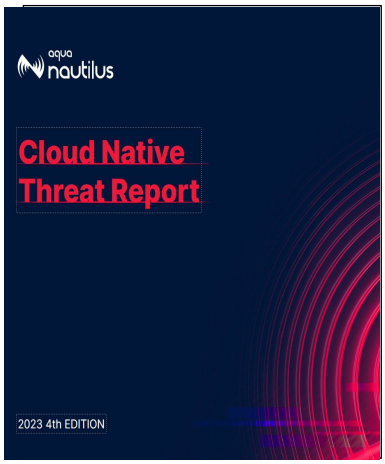


Cloud Native Threat Report

Thank you for downloading this Aqua resource. Carahsoft is the Market Vendor for Aqua cybersecurity solutions available via GSA-70, ITES-SW, CMAS, and other contract vehicles.

To learn how to take the next step toward acquiring Aqua's solutions, please check out the following resources and information:

For additional resources:
carah.io/AquaResources

For upcoming events:
carah.io/AquaEvents

For additional Aqua solutions:
carah.io/AquaSolutions

For additional Cybersecurity solutions:
carah.io/Cybersecurity

To set up a meeting:
aquasec@carahsoft.com
866-421-4683

To purchase, check out the contract vehicles available for procurement:
carah.io/AquaContracts

For more information, contact Carahsoft or our reseller partners:
aquasec@carahsoft.com | 866-421-4683

aqua nautilus

Cloud Native Threat Report

2023 4th EDITION

Table of Contents

Executive Summary

The report below summarizes research and observations compiled over the past year by Aqua Nautilus threat researchers. Based on actual attacks in the wild and analysis of threat actors' changing tactics, techniques, and procedures, the report provides security practitioners greater insight into the mind of the attacker Ultimately, these insights can help you make better, faster decisions to protect your entire cloud native stack.

The report is structured to cover three key cloud native threat areas:

- Software supply chain
- Risk posture & vulnerabilities misconfigurations
- Protecting workloads in runtime

Key findings

Threat actors are using many techniques to conceal their campaigns.

Aggregated honeypot data, over a six-month period, showed that more than 50% of the attacks focused on defense evasion. These attacks included masquerading techniques, such as files executed from /tmp, and obfuscated files or information, such as dynamic loading of code.

In addition, in 5% of the attacks, threat actors used a memory resident malware. Compared with prior Aqua Nautilus research in 2022, **there was a 1,400% increase in fileless attacks**. This clearly indicates that threat actors are now focusing more on ways to avoid detection to establish a stronger foothold in the compromised system.

Runtime security is crucial.

The most persuasive evidence for the threats actors' increasing and successful efforts to evade agentless solutions was found in early 2023. Nautilus researchers discovered HeadCrab, a state-of-the-art, stealthy, Redis-based malware that compromised more than 1,200 servers. The level of its sophistication emphasizes the need for agent-based scanning to detect attacks designed to evade volume-based scanning technologies.

Software supply-chain services can pose inherent risks. Misconfigurations aren't being taken seriously.

Over six months, Aqua Nautilus researchers observed organizations of all sizes at risk for misconfigurations. Specifically, we found more vthan 25,000 distinct servers or smaller organizations that were vulnerable because of misconfigured Docker daemons. On average, each of these servers was exposed for 56 days, or almost two months — ample time for an attacker to find and exploit them.

We also saw this with larger organizations and enterprises. We investigated the issue regarding the kubelet API, and over a six-month period we found that 1,000 servers were exposed. Though the misconfigurations affected fewer servers, on average it required more than 100 days to rectify them.

Even more alarming, API servers allow an attacker greater access to the corresponding cluster. In this case, we found that hundreds of thousands of servers are exposed to the world, with thousands of those servers being potentially exploitable.

Emerging Cloud Native Attacks and Business Risks

> The velocity and scale of the cloud native environment requires you to make better, faster business decisions when prioritizing your limited resources. The complexity, level of experienced and proficient labor, and enlarged attack surface of cloud native environment are the leading edge of business risk.

Cloud computing has revolutionized the way organizations design, develop, deploy, and manage their applications. While this modern approach brings many benefits such as scalability, flexibility, and agility, it also comes with inherent complexities. With the shift to cloud native architectures, the attack surface has expanded significantly, introducing new security risks that must be addressed.

The distributed nature of cloud native applications and their components can make it challenging to identify and mitigate potential risks and threats. It's therefore crucial to learn from the cumulative experiences of your organization and others. There's real value in having some idea of how threat actors might exploit your cloud native environments and the technological core of your business, what to look for, and even anticipate.

Because of the sheer scale and growing diversity of threats against the cloud native tech stack, we've organized this report around three different areas:

- **Threats targeting the software supply chain**
- **Risk posture (vulnerabilities and misconfigurations)**
- **Threats targeting workloads in runtime environments**

About Aqua Nautilus

Aqua Nautilus focuses on cybersecurity research of the cloud native stack. Its mission is to uncover new vulnerabilities, threats, and attacks that target containers, Kubernetes, serverless, and public cloud infrastructure — enabling new methods and tools to address them.

With a global network of honeypots, Aqua Nautilus catches more than 80,000 cloud native attacks every month, specifically those unique to containers and microservices that other platforms cannot see.

For more Nautilus research
aquasec.com/research/

From Code to Cloud: Connecting the Dots in the Cloud Threat Landscape

The creation and distribution of software in a cloud-based environment involves a complex network of dependencies, where multiple parties, including cloud service providers, source-code management applications, CI/CD tools, and registries are involved in various stages of the process. Cloud-based software systems are highly interdependent and consist of multiple layers of components that interact with each other, making it challenging to secure the software supply chain.

This complexity presents a large attack surface that includes various applications, potentially leading to misconfigurations and vulnerabilities. Data from Aqua's software supply-chain team showed that software supply-chain attacks grew by more than 300% year-over-year, while other reports cite an increase of 600% to 800%.

This section highlights the potential impact of software supply chain attacks on businesses and how they can affect other environments in the cloud. By examining real-world examples, businesses can gain a better understanding of emerging areas of risk, prioritize their mitigationre sources and investments, and develop the necessary skills to secure their cloud native environments.

What We Learned from Recent Software Supply Chain Events

This has been an eventful year. There have been multiple, high-profile incidents related to software supply-chain security in the cloud. Let's "shift left" and look first at the popular source-code management platform GitHub.

In April 2022, GitHub's security team reported a security breach that allowed attackers to obtain OAuth user tokens issued to third-party integrators, access data from several GitHub customers, and selectively target specific organizations' private repositories. Not surprisingly, services such GitHub, PyPI, Ruby, and NPM are frequently targeted by threat actors. In September 2022,we saw a new platform deployed for the use of phishing-as-a-service against code and package managers. This tactic bypasses multifactor authentication mechanisms, leading to potential session cookie hijacks and account takeovers.

The severity of this issue was further highlighted by Illustria with the discovery of an account takeover vulnerability in a popular NPM package, affecting over 1,000 organizations. A similar attack affected the Python CTX project. And the PyTorch dependency confusion that occurred later in the year resulted in thousands of developers downloading a malicious binary that exfiltrated data through the DNS.

Such incidents can result in the inadvertent execution of malicious code by a developer or DevOps engineer who relies on external code packages as part of the modern development cycle's dependence on open source software.

Additional evidence supporting the severity of these cases is evidenced in the Python Packages Index maintainers' announcement indicating that they are distributing to over 4K critical projects' maintainers, (as they defined critical), a two-factor authentication key (Titan Security Keys) to fight account takeover.

CI/CD environments have also been an increasingly popular target over the last year. Threat actors use various techniques to exploit popular cloud CI/CD platforms, such as those used in cryptomining, to maximize their gains while maintaining a low footprint.

ORIGINAL NAUTILUS RESEARCH

Supply Chain Threats

One area that Aqua Nautilus focused on is how threat actors can exploit software packages and use them as attack vectors to subvert the software supply chain. Through our research, we demonstrated how attackers can perform reconnaissance and exploit packages in the NPM package manager.

This involved using NPM's API to detect private packages and identifying flaws in two-factor authentication that could enable account takeover attacks.

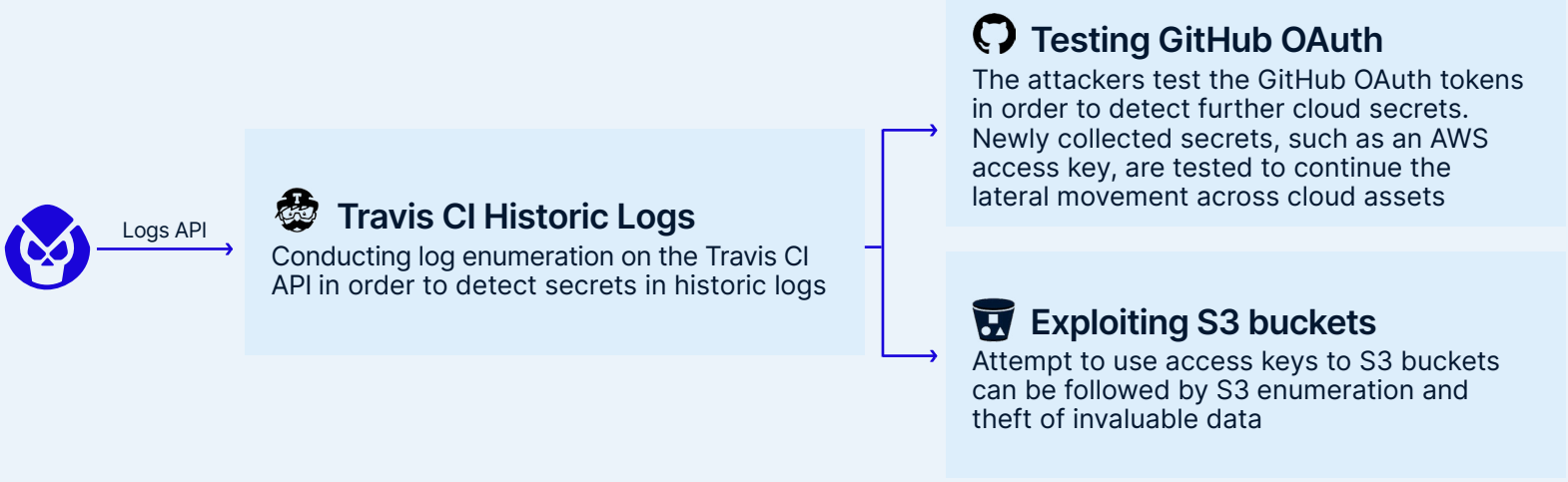
We also discovered a logical flaw called package planting, which allows attackers to disguise malicious packages as legitimate. Additionally, we reported a vulnerability (CVE-2022-32223) in all Node.js versions that could allow the embedding of malicious code into packages and lead to privilege escalation and malware persistence in Windows environments. All findings were reported to the relevant teams and fixed in subsequent security releases.

Sometimes the software supply-chain services that you use offer a specific feature that could put you, and your business, at risk. For example, Travis CI API exposes – in clear text – the logs of its free-tier users.

Aqua Nautilus researchers found that over 770 million logs of free-tier users **were exposed to the internet**. When we downloaded a sample of 7 million logs (~1%), our researchers found that tens of thousands of tokens, secrets, and other credentials were exposed; 50% of these secrets and credentials were still active.

These secrets led to popular cloud service providers such as GitHub, AWS, and Docker Hub. Attackers can use this sensitive data to launch massive cyberattacks and to move laterally in the cloud to other services and private cloud infrastructure.

Simulated lateral movement in the cloud using a compromised token



Over 770M logs of free tier users were **exposed to the internet**

Sample of 7M logs (~1%)

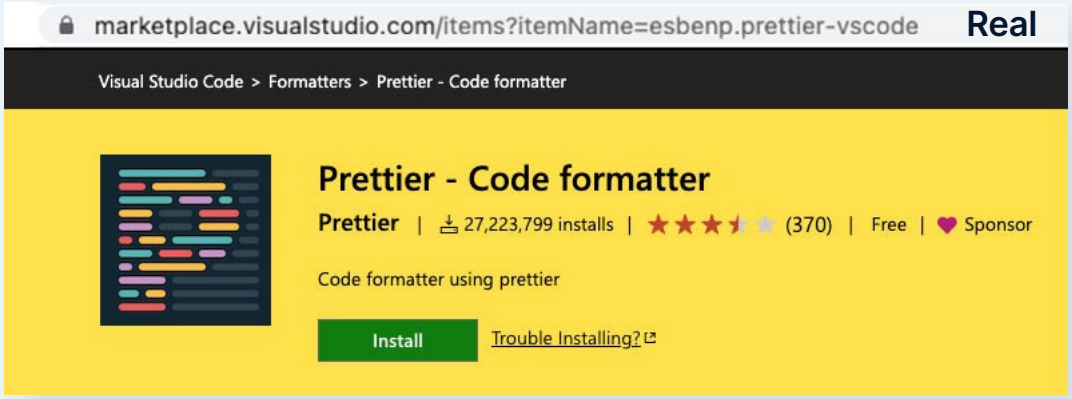
Tens of thousands of tokens, secrets, and other credentials **were exposed**.

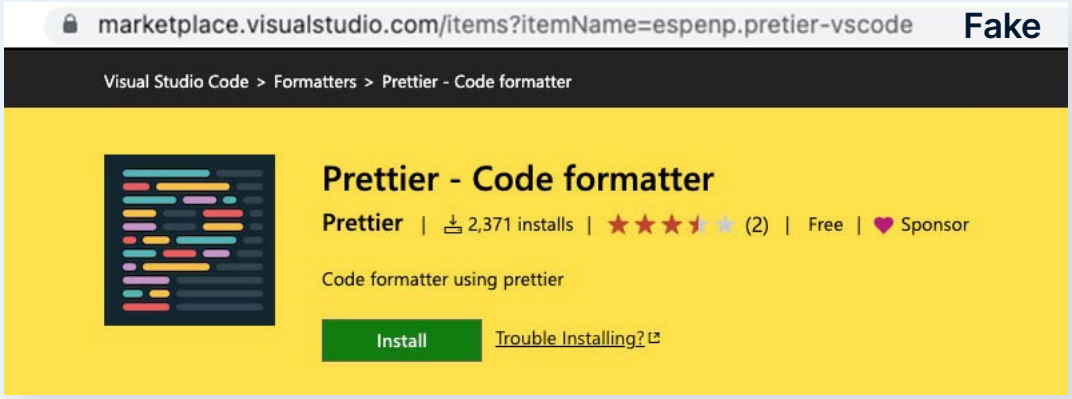
50% of these secrets and credentials **were still active**

One of the most interesting studies that Aqua Nautilus conducted in the supply chain area was related to "shift-left-left," which basically means attacks targeting the developers themselves. For example, research showed how easily developers' trust in common VSCode extensions on the marketplace can be exploited to deliver malware. Considering that approximately 75% of developers use VSCode, we questioned whether blindly trusting marketplace extensions is a best practice.

The research demonstrated that the answer is a resounding no. Aqua Nautilus showed how easy it is for attackers to impersonate popular VSCode extensions and trick unknowing developers into downloading them (see below images). This original research uncovered a new attack method that leverages compromised extensions and could act as an entry point for an attack on many organizations.

Read more on the blog: **Can You Trust Your VSCode Extensions? ›**





Additional recent research by Aqua Nautilus showed how registries, which are a key part of the software supply chain in the cloud, can serve as initial access to expand throughout the cloud. One example illustrates how an international tech giant is at risk due to shadow IT of a single container image registry.

Two misconfigured container image registries that belonged to the development and engineering teams working on a Fortune 100 tech giant's cloud were discovered.

One of the container image manifests contained an active API key to an artifact registry that contained 2,600 repositories with over 240 million artifacts.

The researchers found built artifacts affiliated with the production environment and repositories for managing internal software libraries. Moreover, the API key had "can deploy" privileges, which could allow a bad actor to poison artifacts such as libraries, images, and releases.

Assessing Risk Posture

Robust and common practices in cloud security include finding and patching vulnerabilities and locating and fixing misconfigurations. These are key components in any software security management, let alone in the cloud.

How Insights on Vulnerabilities Protect Your Business

The development of software in the cloud has revolutionized the way businesses operate, but it also continues to introduce new vulnerabilities that threat actors can exploit. These vulnerabilities, coupled with the velocity and scale of cloud environments, require businesses to prioritize their limited resources to make better, faster decisions when addressing vulnerabilities.

This section highlights the potential impact of vulnerabilities on businesses and their resources, data, and revenue streams. Using real-world examples and Aqua Nautilus research, we examine these vulnerabilities so businesses can gain a better understanding of emerging areas of risk, prioritize their mitigation resources and investments, and develop the necessary skills to secure their cloud native environments.

What We Learned from Emerging Vulnerabilities

Vulnerabilities continue to emerge across virtually every stage of the cloud Software Development Life Cycle (SLDC), increasing the exposure and potential risk to your business.

Log4Shell: Rediscovering the Significance of Patching Vulnerabilities

It makes sense to first rewind to December 2021, when Log4Shell (CVE-2021-44228) was seen as a game changer. Many security practitioners described this incident as a race against the clock to detect and patch this vulnerability on their systems.

Threat actors promptly used this vulnerability for attacks in the wild. Just days after the Log4Shell vulnerability was published on December 9, 2021, our honeypots were attacked by large, established botnets such as Muhstik and Mirai and by various smaller endeavors, some of which included advanced techniques such as opening a backdoor to the server with reverse shell attacks. This phenomenon speaks to how bad actors look to repurpose their infrastructure and quickly capitalize on “simple” (and widespread) vulnerabilities.

Later in 2022, the exploitation of the similar X4shell vulnerability raised atttention, but it didn’t have the same impact as Log4Shell. Spring4Shell, in the Spring open source framework for Java applications, and Text4Shell, in the Apache Commons Text Library, caught the media’s attention like Log4Shell. But these applications aren’t as widespread as the Log4j component and were less easy to exploit than Log4Shell. Still, they showed us how vulnerabilities in common tools can quickly affect businesses’ security operations.

Remote Code Execution

There are many flavors to the potential impact of vulnerabilities. Some allow privilege escalation, some expose data, but attackers often look to vulnerabilities that enable remote code execution because these allow them to gain initial access to servers.

We've seen several examples of these over the last year: CVE-2022-41352 in the Zimbra collaboration, CVE-2022-23131 in the Zabbix software tool to monitor IT infrastructure, and CVE-2022-26134 in Confluence, which is a popular collaboration wiki for organizations. These three (out of many other) vulnerabilities have two things in common: They're vulnerabilities in popular, widely used software and, as with Log4Shell, attackers promptly exploited them in the wild. Within days, Aqua Nautilus found that a familiar threat actor, 8220 Gang, was exploiting the Confluence vulnerability to deploy its malware.

Significantly, 8220 Gang was also using various techniques to evade detection and moving laterally to spread the attack across local network and remote hosts

Exploiting Linux Vulnerabilities

Once threat actors gain initial access, they have many techniques in their toolbox. Many of these techniques are aimed at evading detection, bypassing restrictions, and gaining persistence. Several vulnerabilities have been discovered that revolve around the Linux kernel and container runtime and that can allow threat actors to escape containers.

- CVE-2022-0847, aka "Dirty Pipe," enables attackers to modify files, which they might then use to escape from containers.
- Another vulnerability was found recently in the container runtime tool CRI-O (CVE -2022-0811) Essentially, it leverages the passing of parameters to a Kubernetes manifest without sufficient validation and could allow attackers to escape Kubernetes and OpenShift clusters. The runtime interface tool Containerd, specifically its runtime interface plugin for Kubernetes, is vulnerable to arbitrary files from the host being copied into the running container as it launches (CVE-2022-23648).

The Aqua Nautilus 2021 Cloud Native Threat Report, **Attacks in the Wild on the Container Supply Chain and Infrastructure**, reviewed how attackers were escalating their attempts to escape from containers to the host machine. We've seen continued evidence in the wild that attackers are exploiting these vulnerabilities to escape containers and can only assume that they'll continue to exploit these vulnerabilities in the future.

Vulnerabilities anywhere in the software development life cycle have the potential to seriously affect your business, but escaping containers to host systems undercuts the very business promise of cloud native operations.

ORIGINAL NAUTILUS RESEARCH

Risk of Vulnerabilities

Aqua Nautilus was interested in some key junctions in the attack surface of the software development life cycle and conducted focused research on various areas where attackers can target cloud environments. Some of these projects uncovered new vulnerabilities and helped our community to have more secure applications, such as the CVE-2022-32223 in Node.js

The team discovered two vulnerabilities (CVE-2023-27898 and CVE-2023-27905) that show how an unauthenticated attacker can execute cross-site scripting or remote code execution (RCE) on the Jenkins server, perhaps the most popular integration and automation tool in the cloud native environment.

Looking for Vulnerabilities in the Wild

Aqua Nautilus' research also systematically collects intelligence about what kind of vulnerabilities attackers are looking for in the wild. In 2022, scanning by threat actors for the Log4Shell vulnerability was overwhelmingly more common than scans for other vulnerabilities.

This isn't surprising, because the Log4Shell vulnerability was widespread even after patches were released, enabled remote code execution in many environments, and got a lot of media attention.

The top 10 vulnerabilities scanned in 2022 (other than Log4Shell which was overwhelmingly high compared to the rest) were mostly related to the ability to conduct remote code execution. This reinforces the idea that attackers are looking for initial access and to run malicious code on remote systems.

Additionally, we see that Apache servers and services are widely targeted, as Log4Shell, Text4Shell, Spring Framework, and other services are all related to Apache.

Top 10 Vulnerabilities scanned in 2022

1. Log4Shell — **Log4Shell** Server CVE-2021-44228
2. Apache HTTP Server Project — **Apache** HTTP Server CVE-2021-42013
3. Apache HTTP Server Project — **Apache** HTTP Server CVE-2021-41773
4. spring — **Spring Cloud RCE** CVE-2022-22963
5. Text4Shell — **Text4Shell** CVE-2022-42889
6. Cisco — **Cisco** ASA & FTD CVE-2020-3452
7. redis — Lua Sandbox Escape in **Redis** CVE-2022-0543
8. vmware — RCE on **VMware** Identity Manager CVE-2022-22954
9. zimbra — XML XXE on **Zimbra** CVE-2022-22954
10. ORACLE — **Oracle** WebLogic RCE CVE-2022-42889

Severity of vulnerabilities 2022

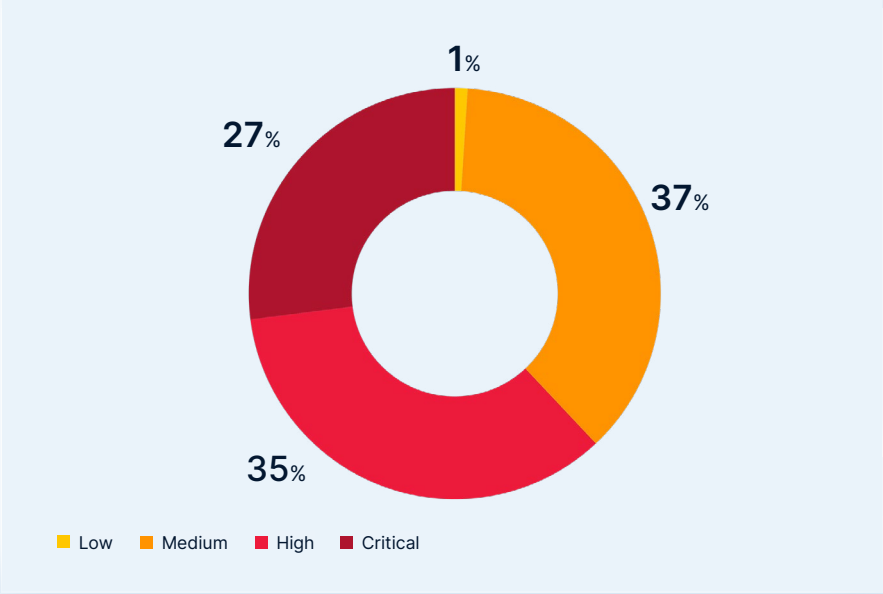


Figure 1 illustrates the distribution of the severity of vulnerabilities (CVSS V3) in 2022, as recorded in attacks against our honeypots. It indicates that attackers try to exploit medium, high, and critical vulnerabilities fairly evenly – likely depending on other factors such as connection to the internet, availability and simplicity of the exploit, and number of potential targets.

Discovery Year for Vulnerabilities

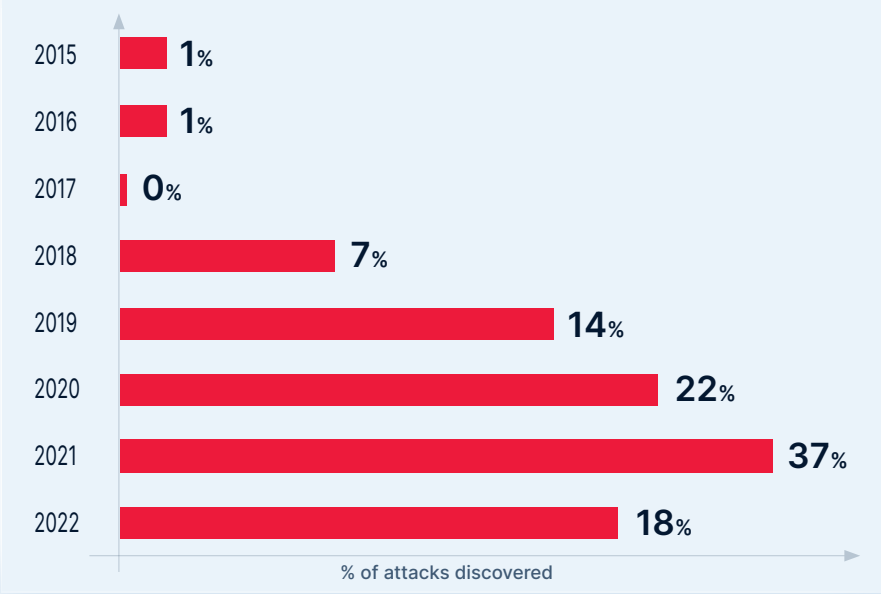


Figure 2 shows the distribution of CVE by publication year, as recorded in attacks against our honeypots. It indicates that the CVEs that were investigated by threat actors in 2022 were discovered in 2021, or even 2020.

This makes sense because it takes time for attackers to detect an exploit, test and arm their infrastructure. But recall as we discovered with the Log4Shell and other vulnerabilities above, we also see many indications that attackers can swiftly add scanning for top critical vulnerabilities to their infrastructure to attack in the wild.

Risks from Misconfigurations: Why You Should Care

Anyone who uses cloud services must take care to properly manage misconfigurations in the cloud because they can cause significant harm to the business. Misconfigurations often arise due to human error, lack of knowledge or experience, or mismanagement of cloud resources.

Simple mistakes could lead to catastrophic consequences, including data breaches, loss of sensitive data, and service disruptions. They can also result in compliance violations and damage to an organization's reputation. Properly configuring cloud infrastructure components and regularly auditing them can help mitigate the risks of cloud misconfigurations.

ORIGINAL NAUTILUS RESEARCH

What We Learned about Misconfigurations

> Over six months, we observed more than 25,000 distinct servers that suffered from a misconfigured Docker daemon. On average, each of these machines was exposed for 56 days, or almost two months — ample time for an attacker using automation to find and exploit them.

We also investigated the misconfiguration issue for kubelet API and found that 1,000 servers were exposed. Though the misconfigurations affected fewer servers, on average it required more than 100 days to rectify them. Even more alarming, API servers allow an attacker greater access to the cluster. In this case, we found that hundreds of thousands of servers were exposed to the world, with thousands of them being potentially exploitable.

Moreover, we found more than 10,000 container image registries exposed to the world. Some of them contained sensitive data that can open room for various attacks against these organizations.

When asked if attackers exploit misconfigurations, Aqua Nautilus can easily answer with a "yes." Through our honeypot infrastructure, we've observed that attackers prefer to exploit misconfigurations rather than vulnerabilities. Typically, a malicious scanner attempts to exploit a poorly configured application or engage in a brute-force attack to guess a weak password (which is a misconfiguration in itself).

Example:

In December 2021, we reported about an investigation of a server that was targeted by the DreamBus botnet due to a poor password. The developer had launched a cloud native application with admin access, but mistakenly configured it with weak credentials. Within just 12 hours, the environment was attacked by the DreamBus botnet, which was able to evade defenses and run Kinsing malware and cryptominers, demonstrating the rapid and severe impact of such exposure.

Monitoring Runtime Is Key

> Protecting workloads in runtime environments is crucial for ensuring the security and integrity of your business data and applications. This is where code executes, and threat actors are increasingly targeting these environments to steal data or disrupt business operations. One might argue that organizations have already hardened their runtime environments and implemented shift-left security practices to address vulnerabilities and misconfigurations in the source code.

So why should they also focus on runtime protection?

There are four very good reasons:

1. It can take time to prioritize and fix known vulnerabilities, which can leave runtime environments exposed.
2. Security practitioners may be unaware of, or miss, supply-chain attack vectors, creating a direct and uncontrolled link to production environments.
3. Even with robust security measures, critical production configurations may still be overlooked in high-velocity, complex, multi-vendor cloud environments.
4. There's always the potential of zero-day vulnerabilities waiting to be exploited, making it essential to have a monitoring system in place for malicious events in production.

Protecting runtime environments requires at least a monitoring approach that includes scanning for known malicious files and network communications, then blocking them and alerting when they appear. However, this is still insufficient. A better solution includes monitoring for indicators or markers that suggest malicious behavior as well – for instance, behaviors such as unauthorized attempts to access sensitive data, attempts to hide processes while elevating privileges, and the opening of backdoors to unknown IP addresses. Ultimately, it's critical to implement robust protection measures in runtime environments to ensure that data and applications are secure and to avoid being vulnerable to attacks. Below we share some of the runtime incidents we encountered over the last year.

ORIGINAL NAUTILUS RESEARCH

What We Learned About Evolving Runtime Security Incidents

Throughout the year, we've seen many new malware and techniques used in various types of attacks. We review some of the attacks and our takeaways from these campaigns.

Recent research designed to locate unknown gaps in security showed that data practitioners, such as data scientists and analysts, were identified as a potential target. Skilled in their craft but often lacking security knowledge and skills, data practitioners are valuable targets because they have the power to access various applications and data environments within the organization. We found that attackers actively seek access to data tools such as Jupyter notebooks and demonstrated many examples.

Three interesting discoveries were made in this realm of data practitioners:

We found the first known **Python**-based ransomware.

Threat actors were deploying **Cobalt Strike**, a powerful attack emulation security tool, to further their attacks.

We found evidence of various vulnerability exploits to elevate privileges.

Taken together, these findings highlight the growing potential risks to this community.

Other findings of original Nautilus research uncovered new and unique methods of hijacking corporate resources, new Linux malware, and possibly a renewed activity by TeamTNT.

Aqua researchers revealed a new type of cryptojacking. While everyone is focused on protecting their CPU resources, we found that attackers were targeting network bandwidth to generate a new type of cryptocurrency.

Towards the end of 2022, Nautilus researchers discovered a new Linux malware , Redigo, a Redis-based malware written in the Golang programming language. This malware had no detections in Virus Total, and more worrisome, it displayed defense evasion techniques.

We've also seen some evidence of renewed activity by TeamTNT with new malware. For knowledgeable cloud security practitioners, this is not good news. Between 2019 and 2021, TeamTNT was quite active, compromising many domains in cloud native businesses. They launched numerous campaigns, sometimes on weekly basis, and introduced many techniques that hadn't been used in cloud native hadn't been used in cloud native attacks before, such as rootkits, fileless malware, obfuscation techniques, and container escape techniques. That said, since this most recent attack, in September 2022, we haven't seen any further evidence that truly supports a return to activity of TeamTNT. Thus, we can conclude that this was a false alarm, and might have been a copycat, someone who used their code, or a one last encore appearance.

In over 50% of the attacks we've seen, attackers used various defense evasion techniques to conceal their attack. There's more behind this figure. Recently we've witnessed the emergence of two highly stealthy, Linux-based malware: BPFdoor and Symbiote

BPFDoor is a passive backdoor used by a China-based threat actor. It supports multiple protocols for communicating with a command-and-control server, including TCP, UDP, and ICMP. This allows the threat actor a variety of mechanisms to interact with the malware – for example, it has used the Berkeley Packet Filtering (BPF) technology in the Linux kernel to evade detection by most firewalls and detection systems.

Symbiote uses various, sophisticated techniques to conceal its operations from most security monitoring and software. These include hijacking system functions and replacing them with its own algorithms, hiding network activity by discovering hidden ports and IP addresses, checking network packets, and changing the BPF filter. Symbiote also steals user credentials by using keylogger functions and can then send the stolen data over DNS.

When considering another trend seen in 2022, involving the adoption of cloud instances volume scanning, one can wonder if decision makers are aware of lack of these solutions. Don't get us wrong: We see value in this type of security solution. Nevertheless, these are not holistic security solutions, and threat actors are fully aware of that. They see the security gaps caused by just using agentless technologies without deploying agents to protect workloads in runtime.

These snapshots of runtime workloads completely miss files written in memory; hence, memory resident malware gets the better of these solutions. Furthermore, no one is performing volume-based scanning in real-time scale. It's impractical. It's impossible. Most of the ones we've seen take a window of 24 hours in between scans. So, if a file was written to disk and then deleted, with history wiped out, the agentless technologies will miss it again. Lastly, even if the agentless technology eventually detects the malware, would you like to detect malware in your system 10 hours after it was deployed?

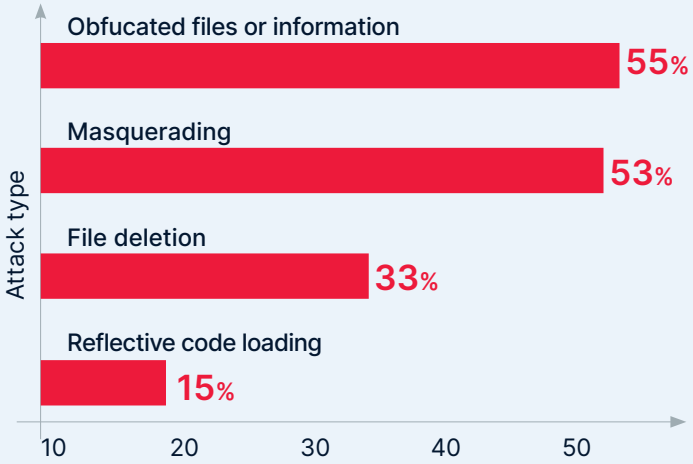
Aqua Nautilus aggregated our honeypot data across all attacks over the last six months of 2022. We clustered all the attacks – over 700,000 – and found that in 63% of them, known malware was detected. This indicates that a security solution that relies solely on antivirus, or solutions that look only for identified malwarme, will fail to detect 37% of these types of attacks.

With the adoption of cloud instances volume scanning, one can wonder if decision makers are aware of lack of these solutions. While we see value in this type of security solution, these are not holistic security solutions and threat actors are fully aware of that, they see the security gaps caused by just using agentless technologies without deploying agents to protect workloads in runtime.

Furthermore as illustrated in the bar chart below, over 50% of all attacks included a masquerading technique, such as a file executed from /tmp, and obfuscated files or information, such as dynamic loading of code.

This is a significant increase compared with a similar analysis done by Aqua Nautilus in 2021, when dynamic code loading was almost 15% less frequent. Also of note, compared with prior Nautilus research in 2022, **there was a 1,400% increase in fileless attacks.**

Various techniques observed in the wild



The HeadCrab campaign

The most persuasive evidence for the threat actors' increasing, and successful, efforts to evade agentless technology was found in early 2023.

Aqua Nautilus researchers discovered a new elusive and severe threat that has been infiltrating and residing on servers worldwide since early September 2021. Known as HeadCrab, this advanced threat actor uses state-of-the-art, custom-made malware that is undetectable by agentless and traditional **antivirus** technologies. Aqua Nautilus found evidence that HeadCrab has taken control of at least 1,200 Redis servers, some of them belonging to security companies.

HeadCrab created his own Redis commands:

```
if ( (unsigned int)RedisModule_CreateCommand(a1, "rdsa", sub_8B94, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdss", sub_9CEA, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdsp", sub_8211, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdsi", sub_9E47, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdsc", sub_12055, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdsm", sub_903B, v7, 1LL, 1LL, 1LL) == 1
    || (unsigned int)RedisModule_CreateCommand(a1, "rdsr", sub_11EE7, v7, 1LL, 1LL, 1LL) == 1 )
{
  return dword_2467FC != 0;
}
unlink("/var/lib/redis/cmd.log");
((void (__fastcall *)(const char *))sub_8F46)("/var/lib/redis/cmd.log");
RedisModule_CreateCommand(a1, "rdsx", sub_C5AC, v7, 1LL, 1LL, 1LL);
return 0LL;
```

Redis is an open-source, in-memory database. Nautilus researchers discovered a new state-of-the-art, elusive malware that is built on the Redis database framework to evade detection by agentless and traditional antivirus solutions.

The malware uses Redis commands and creates new commands to increase capabilities on its victims' servers. The HeadCrab botnet has taken control of at least 1,200 servers.

Conclusions

This report shed light on some of the most significant cloud native threats: software supply chain, risk posture (vulnerabilities and misconfigurations), and protecting workloads in runtime.

What Aqua Nautilus has seen is...

There is a clear indication that threat actors are now focusing more on ways to avoid detection to establish a stronger foothold in the compromised system.

Even a minor misconfiguration in various areas in the software supply chain in the cloud can lead to wide areas of compromise in the software development life cycle.

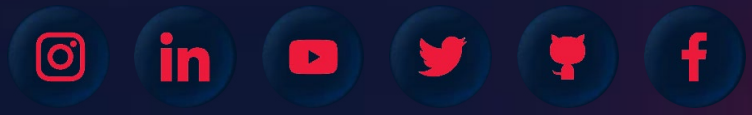
Runtime security is crucial as more threats are targeting workloads in runtime environments, and successful efforts to evade agentless solutions continue to be found in 2023.

Organizations of all sizes are at risk for misconfigurations, and unfortunately aren't being taken seriously.

About Nautilus

Aqua Nautilus focuses on cybersecurity research of the cloud native stack. Its mission is to uncover new vulnerabilities, threats, and attacks that target containers, Kubernetes, serverless, and public cloud infrastructure — enabling new methods and tools to address them.

With a global network of honeypots, Aqua Nautilus catches more than 80,000 cloud native attacks every month, specifically those unique to containers and microservices that other platforms cannot see.



RESEARCHTR-V1JUN2023-EN