paloalto NETWORKS | UNIT 42 BY PALO ALTO NETWORKS

CLOUD

2023

# CLOUD THREAT REPORT

## NAVIGATING THE EXPANDING ATTACK SURFACE

VOLUME 7

# Table of Contents

# Foreword

After two decades of rapid cloud adoption by organizations, 2023 could be considered a turning point for cloud security. The rate of cloud migration shows no sign of slowing down—from $370 billion in 2021, with predictions to reach $830 billion in 2025[1]—with many cloud-native applications and architectures already having had time to mature. The dynamic nature of cloud technology—with feature updates in public cloud services, new attack methods, and the widespread use of open-source code—is now driving awareness of the risks inherent to modern, cloud-native development.

The more organizations that adopt cloud-native technologies, the higher the number of cloud-native applications becomes. The popularity and complexity of the technology then expands the attack surface with vulnerabilities and misconfigurations for cybercriminals to exploit.

Based on data collected over the past 12 months, the *Unit 42 Cloud Threat Report, Volume 7*, provides a wide-angle view of the status of common misalignments leaving the door open to malicious activity. Our team of researchers has analyzed data from a range of sources to identify the most pressing threats facing organizations today. We have also provided practical cyber hygiene recommendations for mitigating these risks and protecting your organization from harm.

Increased awareness will shape the future of cloud security and likely include consolidation of security tools, investment in processes and personnel, adoption of security best practices, and collaboration between organizations and cloud providers to improve security.

We hope this report will serve as an indispensable resource for security professionals and decision-makers, as well as anyone else facing the challenges of cloud security today. We believe that staying educated and proactively addressing potential threats can create a safer and more secure environment for everyone.

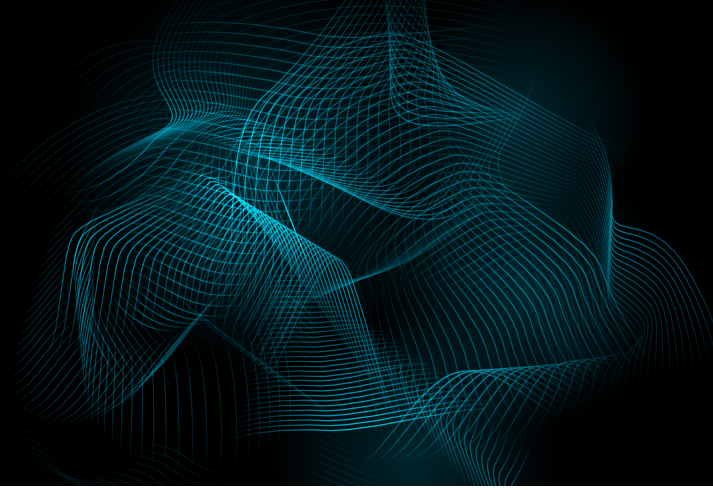
**Ankur Shah**

**Senior Vice President**
**Prisma Cloud**

**Palo Alto Networks**

1. *Cloud Computing Market*, MarketsandMarkets, March 6, 2023, https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html#tab_default_2.

# Executive Summary



To provide security leaders and practitioners with a multifaceted view of cloud security, Unit 42's Cloud Threat Report takes a comprehensive look at the current cloud security landscape using the large-scale data collected in 2022. We examine real breaches that impacted medium and large-size companies, detail the issues observed in thousands of multicloud environments, and analyze the impact of open-source software (OSS) vulnerabilities on the cloud.

Public clouds offer the agility, scalability, and security that on-premises data centers cannot match. In particular, their modern security features can help more effectively secure cloud workloads if implemented correctly. However, the fast evolution and growth of cloud workloads—as well as the complexity of managing hybrid and multicloud environments—cause many organizations to fall behind the curve and inadvertently introduce security weaknesses into their environments, as evidenced by the many legacy resources, vulnerabilities, and insecure configurations we've witnessed. These gaps give adversaries significant opportunities to gain a foothold in the cloud.

This study revealed the following:

**Cloud users repeatedly make the same mistakes**. In most organizations' cloud environments, **5%** of the security rules trigger **80%** of the alerts. In other words, every organization has a small set of risky behaviors that are repeatedly observed in their cloud workloads, such as unrestricted firewall policies, exposed databases, and unenforced multifactor authentication (MFA). Prioritizing the remediation of these issues can maximize the return on security investments.

**Security alerts commonly take days to resolve**. On average, security teams take **145 hours** (approximately 6 days) to resolve a security alert. Well over half **(60%)** of organizations take longer than four days to resolve security issues. Since our previous research showed that it only takes a few hours for threat actors to start exploiting a newly disclosed vulnerability, the current average time to remediate an alert provides a lengthy window of opportunity for potential adversaries.

**Sensitive data in the cloud poses hidden risks**. Sensitive data, such as personally identifiable information (PII), financial records, or intellectual property, are found in **66%** of storage buckets and **63%** of publicly exposed storage buckets. This sensitive data is at risk for both insider and external threats. The lack of insight into what type of information, such as PII or credit card numbers, is stored in each data object makes it difficult to protect sensitive information from being accidentally leaked.

**Leaked credentials in source code are pervasive across all organizations**. The vast majority **(83%)** of organizations have hard-coded credentials in their source control management systems, and **85%** have hard-coded credentials in virtual machines' user data. Leaked credentials are also central to every cloud breach we analyzed. Credential access continues to be a common tactic across all cloud threat actors, and is the approach attackers take to move laterally or vertically in every major cloud breach.

**MFA is not enforced for cloud users**. At least three-quarters **(76%)** of organizations don't enforce MFA for console users, and **58%** of organizations don't enforce MFA for root/admin users. CSPs provide web consoles for users to manage cloud resources visually. However, console access is more susceptible to brute-force attacks as adversaries constantly poke at login pages using leaked usernames and passwords found on the dark web.

**Attacks on software supply chains are on the rise**. The prevalence of open-source usage and the complexity of software dependency make securing the software supply chain difficult. More than 7,300 malicious OSS packages were discovered in 2022 across all major package manager registries according to the GitHub Advisory Database. The impact of these types of attacks is far-reaching. Supply chain attack research showed that techniques such as dependency confusion successfully infiltrated multiple tech giants, and the recent attack that stole GitHub OAuth tokens impacted dozens of organizations. The outcome could be catastrophic if malicious code were committed to these compromised repositories.

**Managing code dependencies is challenging**. Just over half **(51%)** of codebases depend on more than **100** open-source packages. However, only **23%** of the packages are directly imported by the developers. More than three-quarters **(77%)** of the required packages and vulnerabilities are introduced by non-root packages, defined as the dependencies of the directly imported packages. For instance, a developer may import package A to a project, but package A depends on package B and package C. Packages B and C are considered non-root dependencies.

**Unpatched vulnerabilities continue to be low-hanging fruit for attacks**. Nearly two-thirds **(63%)** of the codebases in production have unpatched vulnerabilities rated High or Critical (CVSS >= 7.0), and **11%** of the hosts exposed in public clouds have High or Critical vulnerabilities. In a cloud environment, a single vulnerability in the source code can be replicated to multiple workloads, posing risks to the entire cloud infrastructure.

# Cloud Threats in the Wild

## Cloud Breach Incidents

This section details the tactics, techniques, and procedures (TTPs) that were witnessed in cloud breaches that the Unit 42 Incident Response team handled. The described scenarios have been anonymized and de-identified.

Due to the growth of cloud adoption, Unit 42 has seen an increasing number of cloud breaches. Traditional digital forensics and incident response (DFIR) techniques are not designed to handle these types of events because the tooling, processes, and data sources necessary for investigating security incidents are very different between on-premises and cloud environments.

### Attack Scenario 1: From SIM-Swap to Data Leak on the Dark Web

Bob was a DevOps engineer at a financial firm. One day he was unable to log in to his company's source code management (SCM) system. Thinking he forgot the password, he clicked the "forgot my password" button. While trying to retrieve the password reset link the SCM system sent, he noticed that he couldn't log in to his email account either. Losing access to two critical assets at the same time raised the alarm. He quickly checked his other accounts and discovered that his phone had lost cellular connectivity. Bob had been a victim of a SIM-swap attack. His email and SCM accounts linked to the phone number were successively compromised.

A SIM-swap scam is a mobile phone account takeover fraud that targets two-factor authentication using SMS or phone calls. It happens when an attacker uses social engineering to trick the victim's mobile carrier into activating a new SIM card that the attacker controls. This enables the attacker to take over any victim's accounts that are authenticated through the phone number.

## TTPs

We break down the attack into TTPs following the MITRE ATT&CK Cloud Matrix. The order of the tactics also corresponds to the path of the attack:

- **Initial Access**—how the adversary gains the initial foothold within the cloud:
  - The threat actor SIM-swapped the victim and gained control of the victim's email and SCM accounts linked to the phone number.
  - Next, the threat actor **cloned 600 source code repositories** and uncovered **10 access keys** belonging to four different cloud accounts.
- **Execution**—how the adversary runs the malicious commands:
  - The compromised access keys allowed the threat actor to use the control plane application programming interfaces (APIs) to carry out the rest of the attack.
- **Persistence**—how the adversary keeps access within the compromised environment:
  - One access key with the IAMFullAccess role allowed the threat actor to create new users in the compromised account. Two new users were created to impersonate valid employees.
- **Privilege Escalation**—how the adversary gains more permissions to access more resources:
  - The newly created users were granted more privileged permissions that allowed attackers to perform reconnaissance and move laterally with ease.
- **Discovery**—how the adversary gains more knowledge about the compromised environment:
  - The threat actor enumerated all the virtual machines (VMs), database tables, and storage buckets.
- **Exfiltration**—how the adversary steals data from the compromised cloud workloads:
  - The threat actor:
    - Cloned all the source code repositories.
    - Dumped a subset of database tables and storage buckets that contained sensitive information.

**83%** of the organizations have hard-coded credentials in their source control management systems.

From CTR Vol. 6, **99%** of the cloud identities are overly permissive.

**66%** of the cloud storage buckets contain sensitive information.

- **Impact**—how the adversary manipulates, interrupts, or destroys the compromised cloud workloads:
  - The threat actor:
    - Dropped a subset of tables and storage buckets.
    - Sent the victim a ransom note and threatened to leak the data if the ransom was not paid, a type of attack Unit 42 classifies as Extortion without Encryption. The victim refused to pay the ransom.
  - Some of the exfiltrated data showed up on the dark web a few months later.

The median ransomware demand in 2022 was **$650,000**.[2]

**61%** of cloud accounts have storage buckets that don't enable access logging.

## Key Issues

- **Overly permissive identity**: A DevOps engineer doesn't need access to the entire company's source code repositories, especially when inadequate IAM is one of the top continuous integration/continuous delivery (CI/CD) security risks.[3]

  Privileged permissions such as creating users, editing permission policies, or deleting backups should be granted cautiously. There are also cloud-native and third-party tools that can help down-scope privileges based on the usage history. Granting least-privilege permissions is the most effective way to minimize the impact of security incidents.
- **Credential leak**: Credentials should never be committed to source code repositories, whether publicly accessible or not. Instead, use temporary credential services like AWS STS to dynamically materialize credentials or secret management services like HashiCorp Vault to dynamically provision secrets.
- **Logging not enabled**: The company did not have sufficient logging in place to reliably determine the scale of the data leak. The logs must be kept in locations isolated from the production environment and not accessible by engineering teams to ensure the strongest security.

2. *2023 Unit 42 Ransomware and Extortion Threat Report*, Unit 42, March 21, 2023, https://start.paloaltonetworks.com/2023-unit42-ransomware-extortion-report.
3. Daniel Krivelevich and Omer Gil, *Top 10 CI/CD Security Risks*, Cider Security, last accessed March 10, 2023, https://www.cidersecurity.io/wp-content/uploads/2023/01/Top-10-CI_CD-Risks-OWASP-22.pdf.

## Attack Scenario 2: From Misconfigured Firewall to Cryptojacking Botnet

On a Saturday night, the IT department of a medium-sized e-commerce company received an email from their cloud service provider alerting them of botnet activities in their cloud infrastructure. Because no production workloads were impacted and the issue didn't seem to be urgent, the issue was not escalated further. When the IT administrator, Alice, logged in to the cloud console on Monday morning, she noticed a spike in the monthly bill. A further investigation showed that hundreds of unknown VM instances had been created across multiple regions the previous Friday, all of which were incurring high GPU usage and network traffic.

The company had been a victim of a cryptojacking attack where the threat actor deployed hundreds of VM instances to perform large-scale cryptomining and botnet operations.

### TTPs

- **Initial Access**
  - An internal web server was accidentally made public due to a misconfigured security group setting during a migration process.
  - The threat actor found the exposed host and confirmed the web server was vulnerable to CVE-2021-40438. This Server Side Request Forgery (SSRF) vulnerability allowed attackers to send HTTP requests to hosts behind the firewall.
  - Coupled with the usage of outdated **Instance Metadata Service Version 1 (IMDSv1)**, the SSRF vulnerability allowed the threat actor to exfiltrate temporary credentials associated with the VM instance.
- **Execution**
  - The obtained cloud credentials enabled the threat actor to use control plane APIs to carry out the rest of the attack.
  - The threat actor used infrastructure as code (IaC) service CloudFormation to deploy resources needed for performing cryptojacking (e.g., VPC, subnet, firewalls, and VM instances).
- **Persistence**
  - The threat actor created a backdoor IAM role that allowed access from an attacker-controlled account.

**CVE-2021-40438** is one of the top 10 Common Vulnerabilities and Exposures (CVEs) identified in exposed cloud hosts.


**55%** of organizations still have AWS Elastic Compute Cloud (EC2) instances configured with IMDSv1.

**35%** of cloud accounts have cross-account resource access through IAM.

- **Lateral movement and privilege escalation**
  - Control plane → data plane:
    - The threat actor enumerated and viewed all the VM instances' user data. A **hard-coded GitHub credential** in one VM instance's user data allowed the threat actor to access the source code repositories for the entire company.
  - Data plane → control plane (privilege escalation):
    - The threat actor cloned all the repositories and scanned for more credentials. A cloud access key hard-coded in a CI/CD automation script granted the attacker administrator permission.
- **Exfiltration**
  - The threat actor cloned all the repositories in the company's GitHub organization.
- **Impact**
  - Financial resource exhaustion:
    - The threat actor deployed hundreds of GPU-based VM instances, costing the company **$12,000 daily**.
  - Resource abuse:
    - The threat actor deployed botnet malware on the compromised hosts and launched **DDoS attacks** from the company's network.
    - If the incident was not handled properly, the cloud service provider would have suspended the company's accounts and caused an even larger disruption.

## Key Issues

- **Overly permissive network access setting**: Public inbound firewall rules/security groups (0.0.0.0/0) pose hidden risks. Such rules should be flagged and prevented from associating with services not intended to be public.
- **Ineffective vulnerability management**: The company failed to identify and patch critical vulnerabilities in its cloud workloads.
- **Use of outdated cloud service**: IMDSv1 lacks the richer security features available in IMDSv2. IMDSv2 would have prevented the threat actor from exploiting the SSRF vulnerability to gain access tokens.

**85%** of organizations have hard-coded credentials in virtual machines' user data.

From *CTR Vol. 6*, administrator access is among the **top three most used roles.**

**75%** of organizations have VM instances with non-HTTP ports exposed to the public internet.

**11%** of hosts exposed in public clouds have **high or critical vulnerabilities**.

- **Credential leak**: Credentials should never be hard-coded in plaintext in the source code or configurations.
- **Overly permissive identity**: Administrator access should never be granted to any service. Every identity's permissions should be tailored according to the actions it actually performs.

## Cloud Threat Actors

In the Cloud Threat Report, Vol. 6, Unit 42 announced the first Cloud Threat Actor Index to assist security operation teams, threat hunters, researchers, and intelligence professionals in tracking threat actors who target cloud infrastructure. The data contained within the index follows the MITRE ATT&CK cloud and containers matrices, giving security professionals a common framework around which to communicate and discuss the TTPs employed by threat actors. The index also employs the Unit 42 ATOM service to provide security professionals with all of the known indicators of compromise (IoCs) used by threat actors packaged within the industry standard STIX/TAXII format.

Unit 42 researchers define a cloud threat actor as "an individual or group posing a threat to organizations through directed and sustained access to their cloud platform resources, services, or its embedded metadata."

This section provides an overview of new cloud threat actors as well as new activities since the last report. Please refer to the Appendix section for the complete TTP matrices.

| Threat Actor | Operation Updates | Unit 42 ATOM |
|---|---|---|
| **PurpleUrchin (New in 2022)** | PurpleUrchin is a South African-based threat actor group that primarily abuses cloud-based CI/CD services such as GitHub, Heroku, and Togglebox to perform their cryptomining operations. The author calls this tactic "Play and Run." Unit 42 researchers identified more than 130,000 threat actor-controlled accounts that participated in the campaign. During the peak of the operations in November 2022, three-to-five GitHub accounts were created every minute. To bypass the bot prevention mechanism CAPTCHA that some service providers deploy, the threat actor developed an image analysis technique that automatically solved the challenges. | **Automated Libra** |
| **Kinsing** | Kinsing was observed to exploit weakly configured PostgreSQL containers and vulnerable images to compromise Kubernetes clusters. Kinsing also removed the syslog files on the compromised systems. Log removal was the first known effort by the actors to erase the traces of their operations. We added the following MITRE techniques to their operations: T1070.004 - Indicator Removal: File Deletion, T1190 - Exploit Public-Facing Application, and T1525 - Implant Internal Image. | **Money Libra** |
| **WatchDog** | WatchDog was observed to use Steganography to mask the transfer of malware that was hosted on compromised cloud storage buckets. This was the first time that this evasion technique was witnessed in the cryptojacking world. We added the following MITRE techniques to their operations: T1584.004, Compromise infrastructure: Servers, T1059.001 - Command and Scripting Interpreter: PowerShell, and T1027 - Obfuscated Files or Information: Steganography. | **Thief Libra** |
| **8220** | 8220 was observed to deploy Tsunami IRC Botnet and remove syslog on compromised hosts. 8220 massively expanded its botnet operations to as many as 30,000 compromised hosts in 2022. We added the following MITRE techniques to their operations: T1070.002 - Indicator Removal: Clear Linux or Mac System Logs, T1070.004 - Indicator Removal: File Deletion, and T1584.005 - Compromise Infrastructure: Botnet. | **Returned Libra** |

# Oversights in the Cloud

This section provides an overview of the most common security issues observed across the cloud environments of more than a thousand organizations. In particular, we analyzed the workloads in 210,000 cloud accounts, subscriptions, and projects over 1,300 organizations across all major CSPs.

While insecure configurations introduced by users are still the primary concern, we also noticed issues stemming from the ready-to-use templates and default configurations provided by CSPs. These settings and features are convenient, making the adoption of new technologies frictionless. At the same time, they don't position users in the most secure initial state.

## Hard-Coded Credentials

- **83%** of organizations have hard-coded credentials in their source control management systems.
- **85%** of organizations have hard-coded credentials in virtual machines' user data.

We often see credentials stored in plaintext or unencrypted form on storage media, such as files, databases, configurations, logs, and images. These "hard-coded" credentials pose significant security risks because adversaries can use them to bypass most of the defense mechanisms. Credential access is also a common tactic that cloud threat actors leverage to move laterally or vertically. Scraping and exploiting hard-coded credentials is a common finding in every major cloud breach and, when coupled with the pervasive issue of overly permissive cloud identities, enables threat actors to get right to the crown jewels with a single key.

**Tip**: Enable secret scanning in SCM systems, such as GitHub, GitLab, and Bitbucket, and enforce policies to prevent code with secrets from being committed. Scan for secrets in compute resources such as containers and VM instances. Simply deleting or blocking the files with leaked credentials is insufficient; security teams should identify and remediate the root causes to prevent the same problem from being created again.

# Weak Authentication

- **76%** of organizations don't enforce MFA for console users.
- **58%** of organizations don't enforce MFA for root/admin users.
- **57%** of organizations don't enforce symbols in passwords.
- Brute-force attacks on cloud consoles are detected in **43%** of organizations.

Due to the open nature of public cloud administrative points, such as graphical web consoles and API gateways, attackers typically do not need to overcome network boundaries to brute-force authentication or abuse stolen credentials. All CSPs offer internet-facing web consoles that allow users to manage cloud resources visually in the browser. Cloud administrators or root users commonly use web consoles to manage their cloud infrastructure. Console users, however, are more susceptible to brute-force attacks as adversaries constantly poke at login pages using leaked usernames and passwords found on the dark web. Weak passwords that are too short or do not contain a variety of characters are easy to crack.

**Tip**: Strong authentication is the first line of defense to keep attackers outside your cloud workloads. Enforce MFA for all console logins and APIs of critical services, such as IAM and key management service. If possible, disable logging in with passwords and use federated authentication such as Okta and Microsoft Active Directory to authenticate with cloud services.

# Disabled Logging

- 75% of organizations don't enforce trail logs for Amazon Web Services (AWS) CloudTrail.
- 74% of organizations don't enforce Microsoft Azure key vault audit logging.
- 81% of organizations don't enforce Google Cloud Platform (GCP) Storage bucket logging.

Disabled logging is a distressingly common issue, even though CSPs generally offer logging capabilities for most cloud-native services. The problem is that due to extra storage and cost, logging is usually disabled by default, inhibiting visibility and making debugging and threat detection difficult. Lack of visibility during a breach incident leads to a longer detection time, a larger blast radius, and a higher remediation cost.

**Tip**: Enable control plane audit logs, such as AWS CloudTrail, Azure Activity Log, and GCP Cloud Audit Logs, for all cloud environments, as well as resource logging for every workload in production environments. All logs should be consolidated in a centralized and protected location where security analysts can query data easily.

# No Automated Backup

- **49%** of organizations don't enforce AWS DynamoDB point-in-time backup.
- **75%** of organizations don't enforce Azure Cloud SQL backup.

Data in the cloud can be intentionally or unintentionally disrupted for various reasons—cyberattacks, human error, software error, or hardware failure. Backup is the last line of defense in a data loss incident, and with the growing ransomware threat, reliable backups are even more crucial.

**Tip**: There should be an automated backup process for any cloud workload that would interrupt business operations if it were to go down. Backups should be stored in protected locations isolated from the production environment across multiple geographic locations to prevent a single point of failure. All organizations should have Business Continuity and Disaster Recovery (BC/DR) plans that incorporate the process of recovering backups.

# Unencrypted Data at Rest

- **55%** of organizations don't enforce AWS EBS volume encryption.
- **44%** of organizations don't enforce Azure SQL encryption.
- **56%** of organizations don't enforce GCP Kubernetes cluster application-layer secrets encryption.

Unencrypted data at rest is a common issue users across CSPs make. Cloud services such as databases, storage, and file systems all support data-at-rest encryption, but they are not always enabled by default, putting the data at risk for unintended exposure, leak, or physical access. In addition, encryption also helps defend against cross-tenant attacks from threat actors who exploit vulnerabilities underlying the cloud infrastructure.

**Tip**: Enable data-at-rest encryption for every cloud resource where data is persisted, including databases, block/object storage, and snapshots. We recommend that organizations import and manage their key materials in the cloud-native key management services to ensure the highest data sovereignty. Rotate encryption periodically to shorten the lifetime of each key and reduce the impact in case of a credential leak incident.

# Inefficient Alert Handling

- On average, it takes **145 hours** (approximately 6 days) for a security alert to be resolved.
- **60%** of organizations take longer than four days to resolve a security alert.
- In most organizations, **5%** of the security rules trigger **80%** of the alerts they receive.

Security monitoring tools can identify issues, but it's usually the users' responsibility to respond to them and resolve them. Given the fact that more than **60%** of organizations take longer than **four days** to resolve security issues, while threat actors typically exploit a misconfiguration or vulnerability within hours, it's clear that there's work to be done to narrow this gap.

Another interesting observation is that most organizations repeatedly make the same mistakes, such as unrestricted firewall policies, exposed databases, and unenforced MFA, all of which likely originate from an isolated number of engineers and IaC templates. These issues vary from organization to organization, but the takeaway is the same for all of them—a small number of repeatable issues drive the largest percentage of problems.

Figure 1 shows the range of time, in days, organizations take to resolve security alerts. Forty percent of organizations resolve their security alerts within four days.

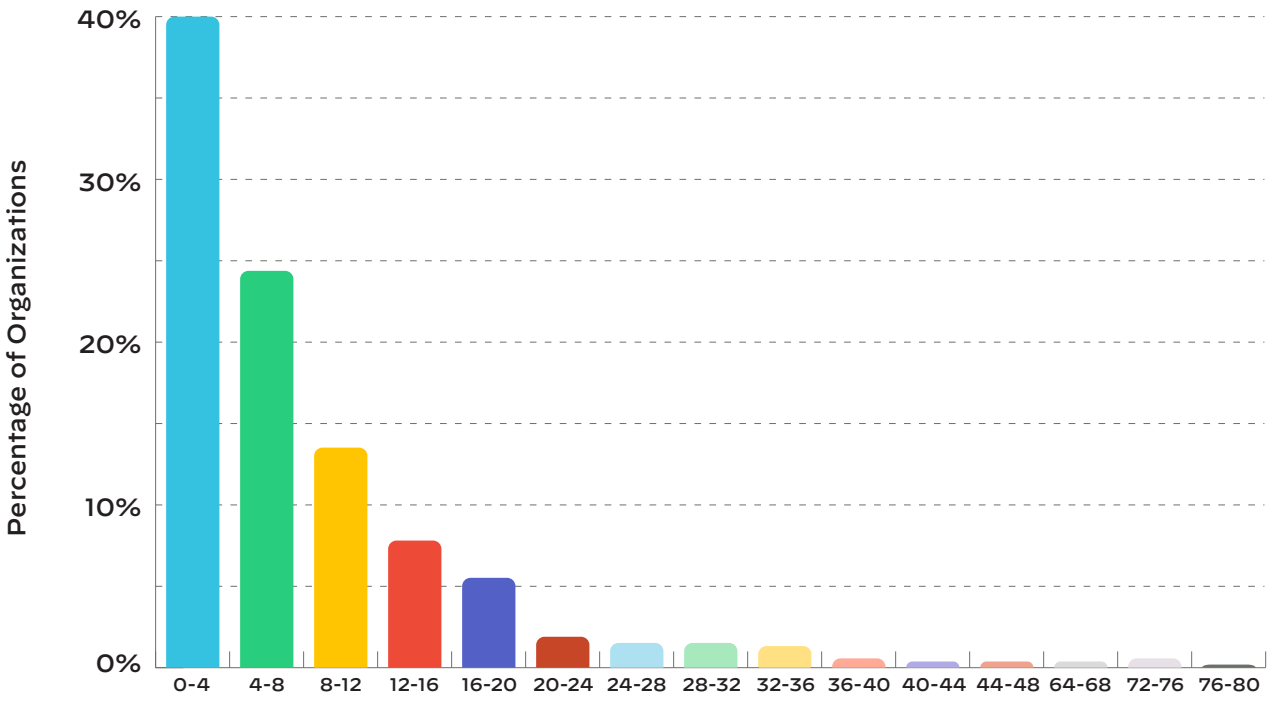



**Figure 1:** Time, in days, organizations take to resolve security alerts

Figure 2 shows the relationship between the number of security alerts an organization receives and the number of unique rules triggering the alerts. The sharp increase in the curve indicates that a small number of the unique rules (x-axis) contributes to the majority of the alerts (y-axis).

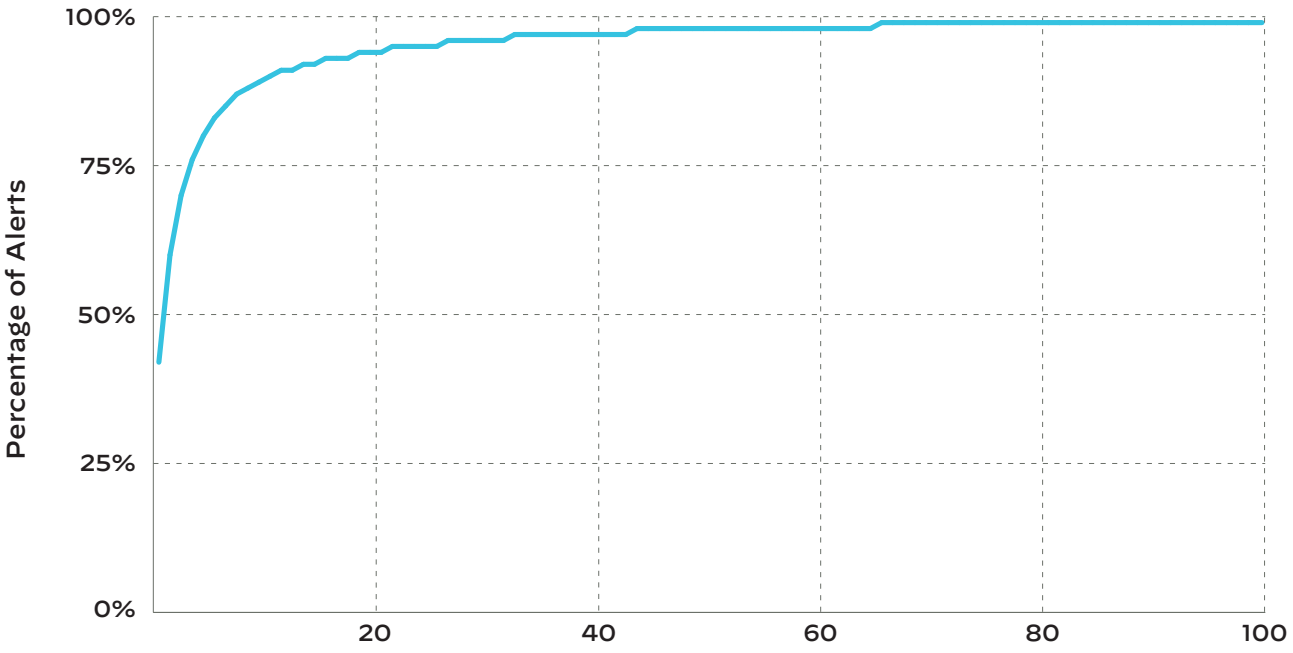



**Figure 2:** Number of unique security rules that contribute to the percentage of alerts organizations receive

**Tip**: Automated remediation and alert triage, as well as shifting security to the left, can all help decrease response times. Organizations should identify their high-frequency alerts and prioritize their remediation strategies. Issues can also be proactively prevented using organization policies (e.g., AWS Service Control Policy, Azure Policy, GCP Organization Policy) or reactively resolved using automated remediation.

# Exposed Sensitive Data

Sensitive data was found to exist in:

- **66%** of cloud storage buckets.
- **63%** of publicly exposed storage buckets.

When managed with care, cloud storage guarantees more security and reliability than on-premises data storage. Cloud storage has become the de facto way organizations back up their business-critical data. However, as cloud applications, services, and users generate data faster than ever, the risk that sensitive information is unknowingly stored at unrestricted locations, exposing it to unintended personnel, increases exponentially. We analyzed data objects in thousands of storage buckets across hundreds of organizations to understand the types of data stored in the cloud. Sensitive data, such as PII, financial records, and intellectual property, were found in many data objects distributed across buckets, including publicly exposed storage buckets.

**Tip**: To prevent these types of issues, it's essential to gain contextual visibility into cloud data. Organizations should adopt data loss prevention (DLP) solutions to continuously identify and monitor the data with sensitive information. Policies should also be created to regulate the retention, access, and protection of sensitive information.

# Publicly Exposed Services

- **41%** of organizations have database services (e.g., SQL Server, MySQL, Redis) exposed to the public internet.
- **73%** of organizations have Remote Desktop Protocol (RDP) exposed to the public internet.
- **75%** of organizations have SSH services exposed to the public internet.

Our research shows that services exposed to the internet are usually scanned and attacked by opportunistic attackers within minutes. Public IP space search engines like Shodan and ZoomEye that continuously look for internet-facing services and make the information freely available to the public also give adversaries an easy pass to millions of potentially unsecured services. Exposed services with unpatched vulnerabilities are also commonly exploited by ransomware actors to gain initial access.

**Tip**: Services such as RDP, databases, and SSH rarely, if ever, need to be exposed to the public internet. Organizations should enforce guardrails to prevent compute instances from being exposed to the public internet using services such as AWS Firewall Manager, Azure Firewall Manager Policy, and GCP Organization Policy.

Figure 3 shows the top 10 internet-facing ports in public clouds. Port numbers have a strong correlation with the services behind the ports because most services have conventional ports that they typically run on. For example, SSH services usually run on port 22 and RDP services on port 3389. Note ports 80 and 443 are excluded from our analysis as web applications are internet-facing by design.

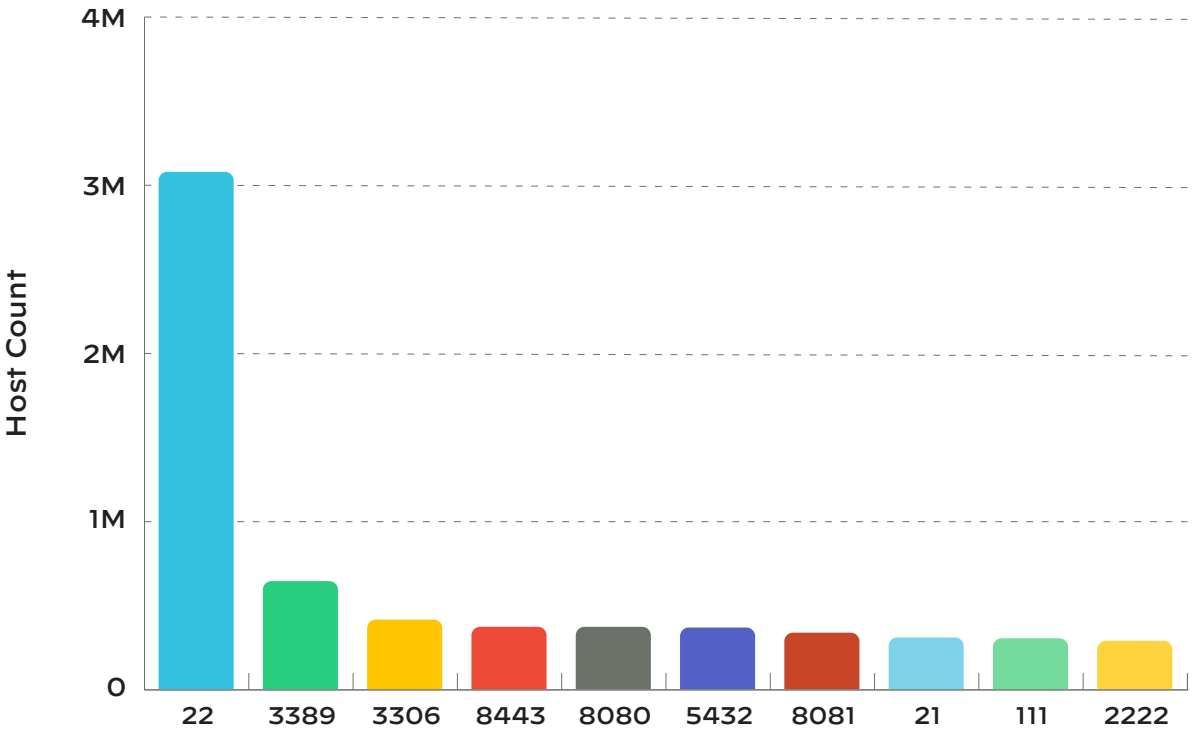



**Figure 3:** Top 10 internet-facing ports in public clouds

# Unpatched Vulnerabilities

Among the **source code repositories** in the production environments, we analyzed:

- **63%** of the repositories have High or Critical vulnerabilities.
- **51%** of the vulnerabilities (High or Critical) are at least **two years old**.

Among the **internet-facing services** that host in public clouds:

- **11%** of exposed hosts contain High or Critical vulnerabilities.
- **71%** of exposed vulnerabilities (High or Critical) are at least **two years old**.

Vulnerability management has always been challenging, but with the popularity of OSS, the scale of what organizations need to manage has grown exponentially. New vulnerabilities can crop up at any time, and a single vulnerability can be propagated to multitudes of cloud workloads due to software dependency. This underscores the fact that no matter how secure the underlying cloud infrastructure is, vulnerable applications in the cloud open up potential attack vectors.

Figure 4 shows the percentage of vulnerabilities categorized by their CVE year. The blue line represents the vulnerabilities found in the internet-facing services in public clouds, and the red line represents the vulnerabilities found in source code management systems. Overall, newer CVEs have a higher chance of being spotted than older CVEs. However, in both cases, more than 50% of the vulnerabilities are older than two years. Note that the exploitability of each vulnerability depends on multiple risk factors, such as configurations and execution paths.

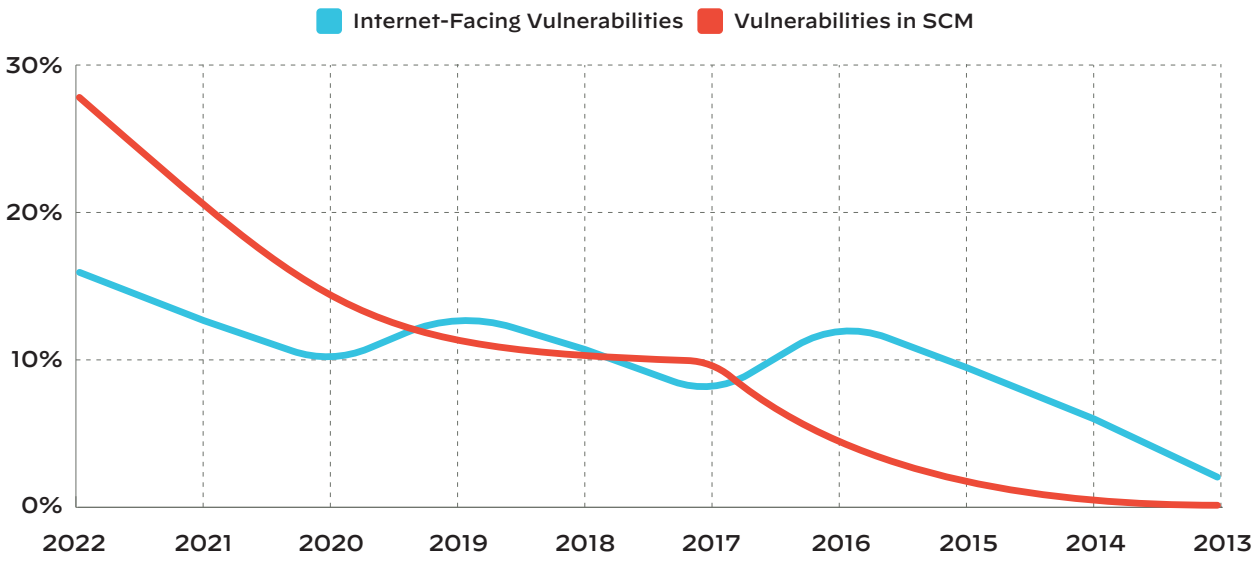



**Figure 4:** Percentage of vulnerabilities categorized by their CVE year

**Tip**: Vulnerability scanning, such as software composition analysis, should be conducted in every stage of the CI/CD pipeline and security policies should be implemented to block code or artifacts with critical vulnerabilities from being deployed.

# Impacts and Risks of Open-Source Software in the Cloud

Open-Source Software (OSS) has been one of the driving forces behind the cloud revolution, with open-source communities such as the Linux Foundation and the Cloud Native Computing Foundation (CNCF) laying the groundwork for modern cloud computing. However, the increased use of OSS could also increase the likelihood of deprecated or abandoned software, malicious content, and slower patching cycles. This puts the onus on end users to scrutinize the OSS before integrating it into applications, a task that's particularly challenging when organizations need to manage scores of projects that are all dependent on potentially thousands of OSS.

## Vulnerability Trends in CNCF Projects

CNCF is part of the nonprofit Linux Foundation that focuses on advancing cloud computing technologies and fostering large cloud-native projects like Kubernetes and Prometheus. We analyzed more than 150 CNCF projects to understand the security posture of these cloud-focused OSSs. CNCF projects are regularly reviewed by the community and the industry alike. There are multiple regular security audits chartered by CNCF which ensure their security robustness.

Figure 5 shows the top eight vulnerability types under CNCF since its inception in 2016, and figure 6 shows the top six programming languages with the most vulnerabilities:

- **Denial of service (DoS)** and **authentication bypass** are the two most common vulnerability types, accounting for **57%** of vulnerabilities.
- The two most popular languages used in CNCF projects are **Go** (61.8%) and **Rust** (8.6%).
- Although only **6.6%** of CNCF projects use C++ as the main language, C++ accounts for **19%** of the vulnerabilities.

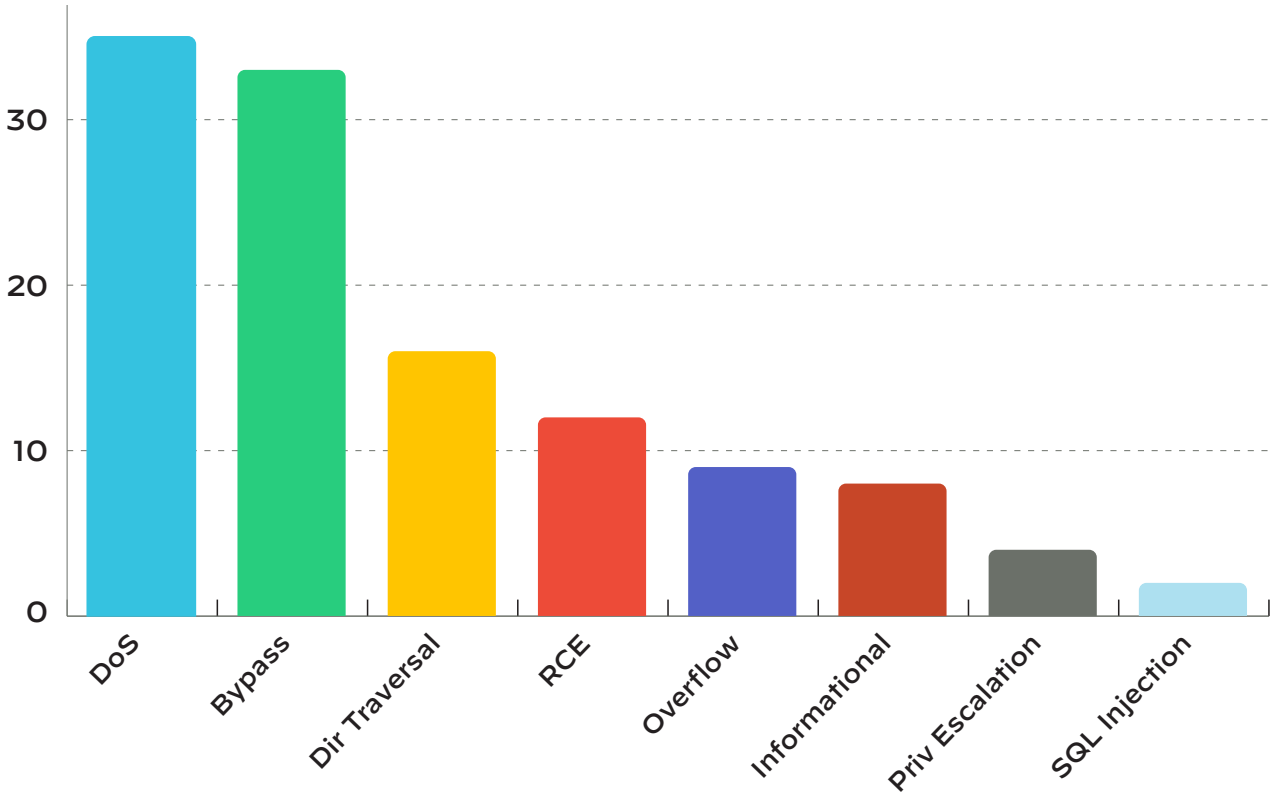



**Figure 5:** Number of vulnerabilities in CNCF projects categorized by vulnerability type

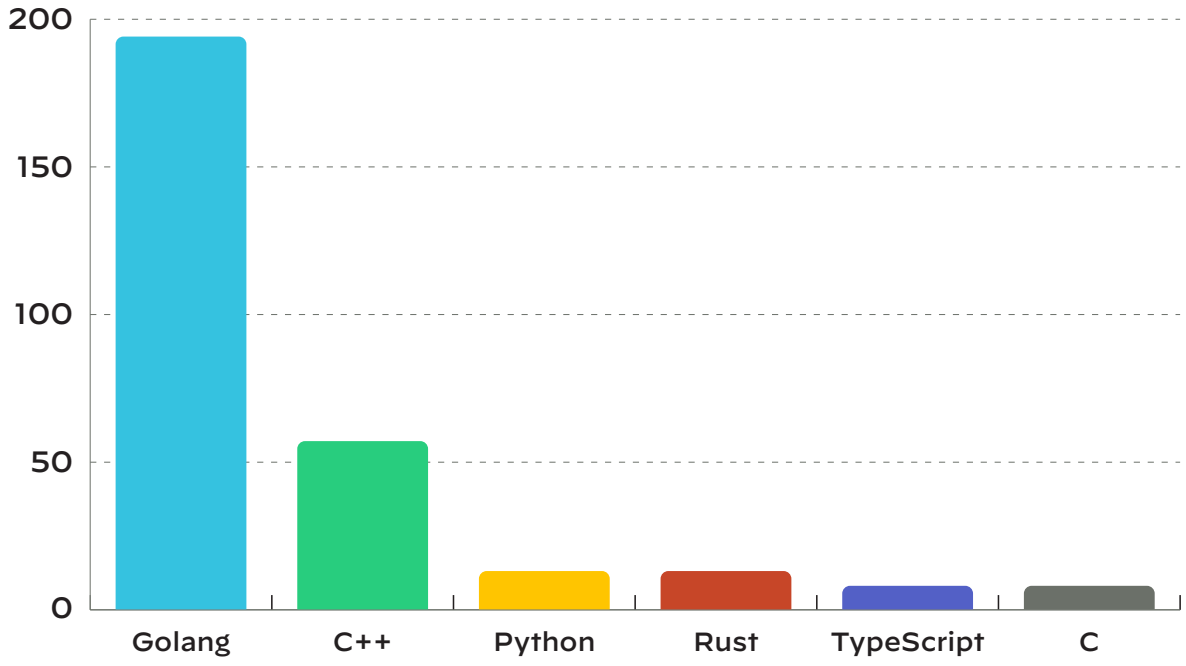



**Figure 6:** Number of vulnerabilities in CNCF projects categorized by language

# Vulnerability Trends in Web Applications

This section provides an overview of the trends of the most disclosed and exploited web vulnerabilities.

## Increasing Web Application Vulnerabilities

We looked into four of the most common types of web application vulnerabilities, **cross-site scripting (XSS)**, **SQL injection**, **cross-site request forgery (CSRF)**, and **directory traversal**. Surprisingly, despite preventive mechanisms such as query parameterization and input/output encoding being built into most modern web application frameworks for years, **SQL injection** and **XSS** are still ranked in the top three most disclosed vulnerability types in 2022.

Figure 7 shows the top 10 Common Weakness Enumeration (CWE) for CVEs disclosed in 2022. The four slices colored in red are vulnerability types most relevant to web or API applications.

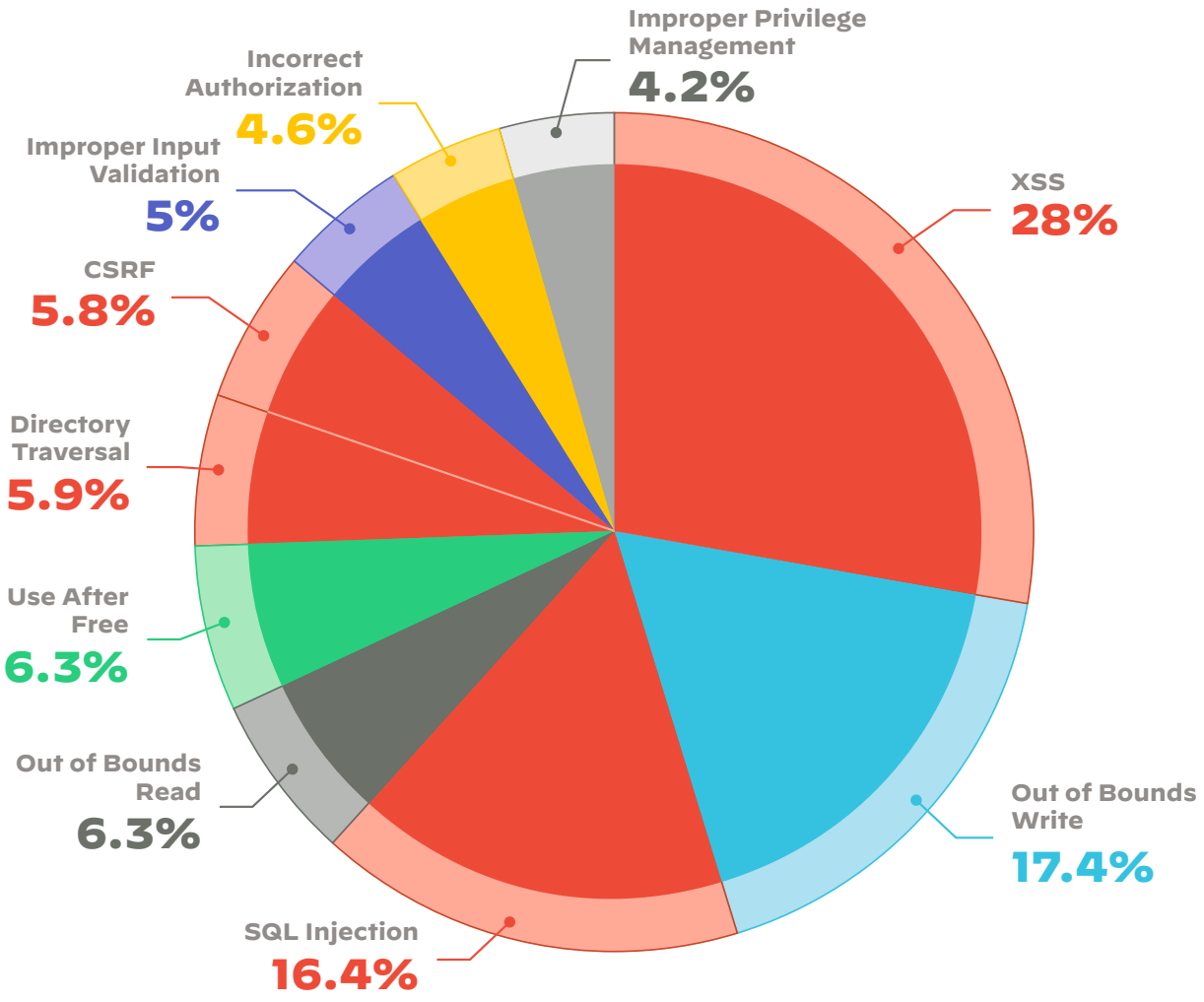



**Figure 7:** Percentage of vulnerabilities categorized by vulnerability type

Figure 8 shows the increasing trends of these four vulnerability types in the past 10 years (note that the base number of systems being examined for vulnerabilities has also grown during this time):

- **9 of the top 10** vulnerabilities on internet-facing cloud hosts belong to web/API applications.
- XSS, SQL injection, CSRF, and directory traversal vulnerabilities account for **54%** of the top 10 vulnerabilities in 2022.
- The growth rate of web-centric vulnerabilities (XSS, SQLI, CSRF, directory traversal) was **1.9 times** faster than the average in 2021.

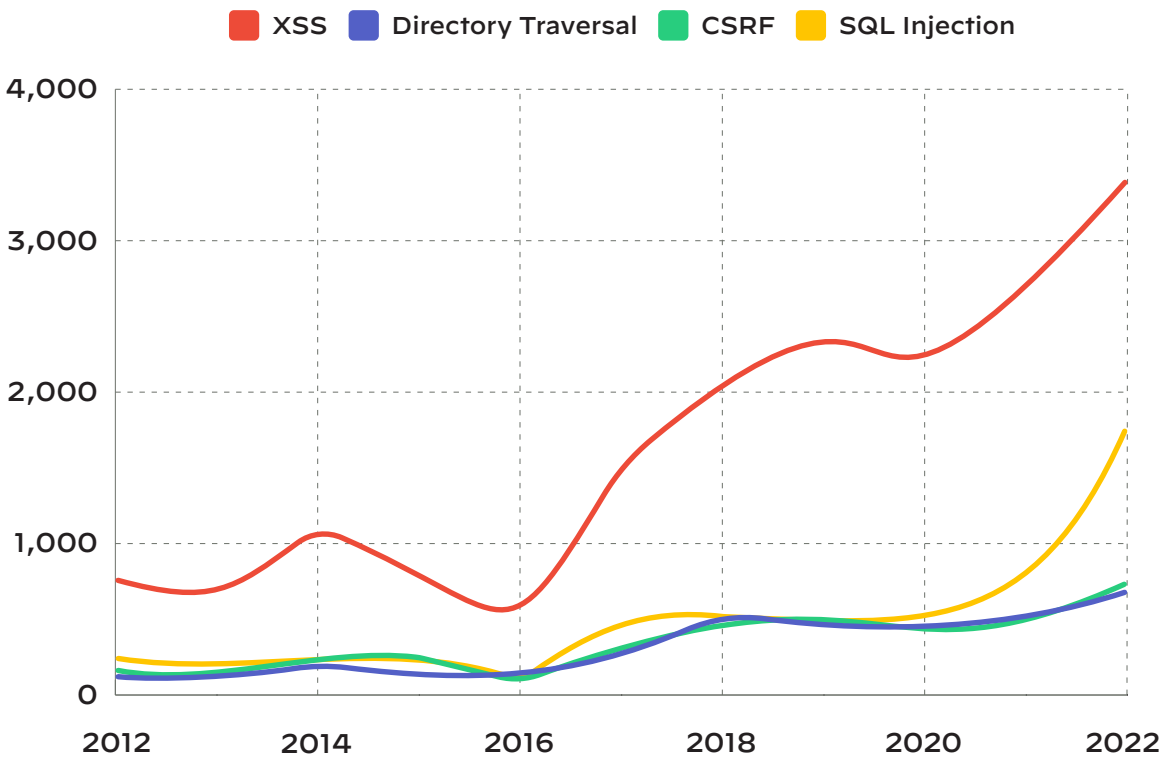



**Figure 8:** Number of XSS, directory traversal, CSRF, and SQL injection vulnerabilities found in the past 10 years

## Increasing Web Application Attacks

Log4Shell (CVE-2021-44228) and Spring4Shell (CVE-2022-22965) were the most notable web application vulnerabilities that adversaries exploited in 2022. These two CVEs are associated with the vulnerable versions of Log4j 2 and Spring Cloud Function. Due to the pervasiveness and ease of exploitation of these two vulnerabilities, their shockwaves impacted almost every organization in every industry. A report published by the Cybersecurity and Infrastructure Security Agency (CISA) Cyber Safety Review Board described Log4Shell as an "endemic vulnerability."

Figure 9 shows the change of Log4Shell exploitation attempts normalized to January 2022, and figure 10 shows the change of Spring4Shell exploitation attempts normalized to May 2022. We believe that publicly available exploits incorporated into botnets and malware accounted for the steady increase in Log4Shell attacks. Even a year after its first disclosure, we still see an increasing trend of exploitation attempts. Note that the number of successful exploitations is unknown to us due to the limitation of the data sources.

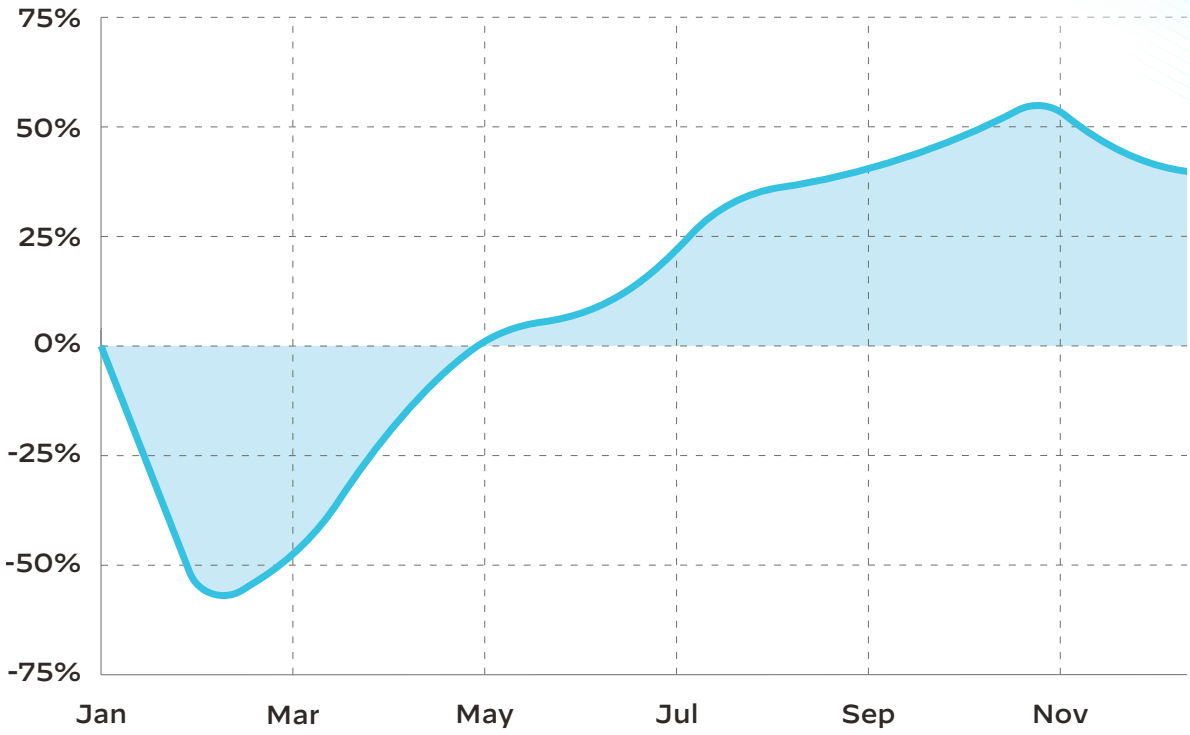



**Figure 9:** Monthly change of Log4Shell exploitation attempts from January 2022 to December 2022

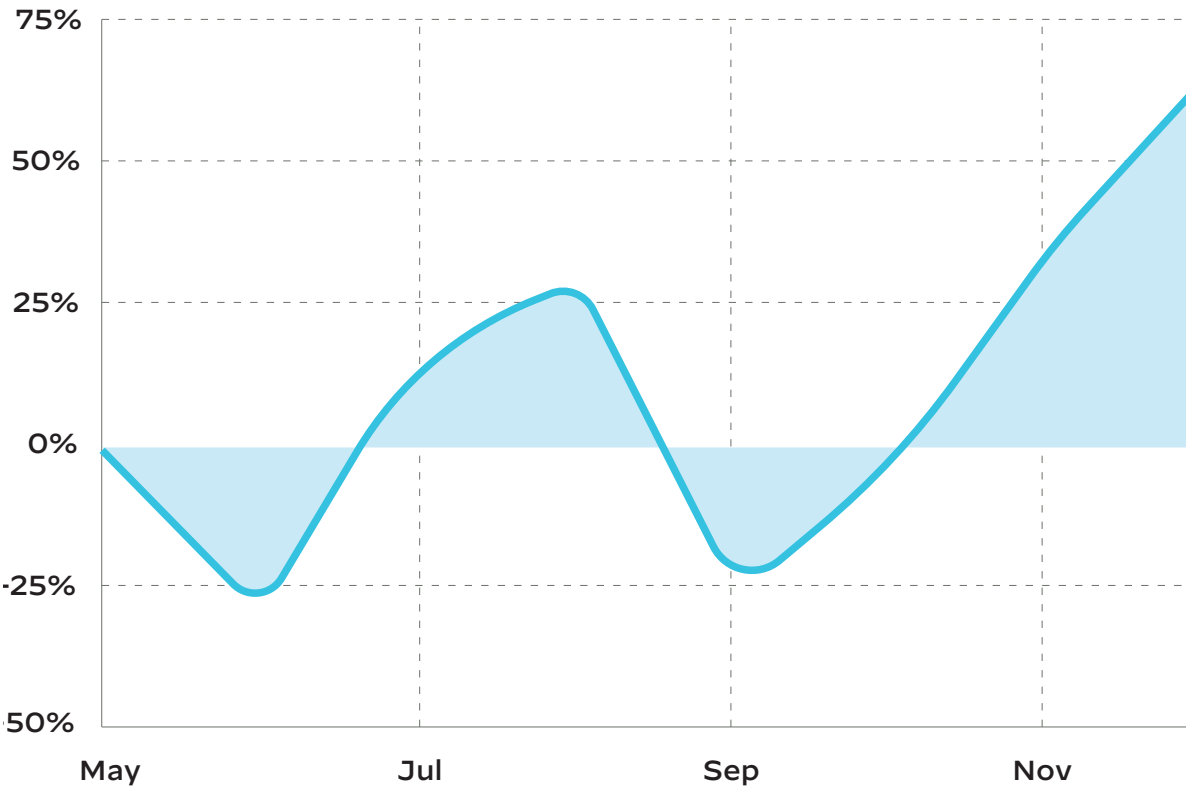



**Figure 10:** Monthly change of Spring4Shell exploitation attempts from May 2022 to December 2022

# Malicious Packages in OSS

This section provides an overview of four common techniques that threat actors use to smuggle malicious content into OSS.

In 2022, more than 7,300 malicious OSS were identified across all major package manager registries, according to the GitHub Advisory Database. Note that the actual impacts caused by these malicious packages are not clear. At the time of writing, the number of total downloads of these malicious packages is unavailable to us as most packages have been removed.

While the number of successful exploits by threat actors is unknown, researchers demonstrated several techniques, such as dependency confusion and account takeover, that effectively infiltrated the software supply chain of multiple large tech companies. Since the potential payout is high and the required cost and skill are low for these kinds of attacks, we anticipate seeing a rise in similar attacks in 2023.

## 1. Typosquatting

Typosquatting is a technique that relies on human mistakes, such as typos, when inputting a string of characters. This technique is not new or unique to malicious packages but has been known for exploiting mistyped URLs. To target an open-source package, an attacker claims many names similar to the targeted package's name on the package manager registry. For example, requests is a very popular Python package with more than 6 million downloads daily. Packages with names such as 'requessts', 'requeests', 'requuests', 'reqquests', 'reequests', and 'rrequests' have been spotted in PyPi. Threat actors also take advantage of major data leak incidents on the news and publish "fake" stolen data that contain malicious code. A recent data leak involving source code exfiltration quickly triggered multiple malicious packages published on NPM. Most of these packages contained malicious code that launched cryptojacking operations or collected sensitive information from the compromised hosts.

## 2. Dependency Confusion

Dependency confusion exploits the lack of distinction between internal and external source code repositories. The technique tricks a build system into pulling a package from an attacker-controlled public repository instead of the privately hosted internal repository. To target an organization, an attacker first identifies the packages that an organization hosts internally. Code in public GitHub, websites, or mobile applications may all reveal such information. The attacker then pushes malicious content to public repositories registered under the same package names. The likelihood is high that a build system on a developer's laptop or CI/CD platform will download the wrong packages if the public repositories have newer versions. More than 5,000 malicious packages built on this technique were discovered less than a month after the original research was published,[4] with more than 4,000 from a single bad actor, all of which were quickly removed.

4. Adam Bannister, "Open source software repositories play 'whack-a-mole' as 'dependency confusion' copycats exceed 5000," The Daily Swig, May 5, 2021, https://portswigger.net/daily-swig/open-source-software-repositories-play-whack-a-mole-as-dependency-confusion-copycats-exceed-5-000.

### 3. Account Takeover

In an account takeover attack, an attacker compromises an OSS package maintainer's account and pushes malicious code to the legitimate repository. This can also happen if an attacker takes over an abandoned or archived repository and pushes malicious content to it. A researcher demonstrated this technique by hijacking two abandoned but still popular packages, 'Python CTX' and 'PHP PHPass'. The researcher achieved account takeover by claiming the expired email domains and the deleted GitHub accounts associated with these projects. The proof of concept packages reached more than 10 million users in just a few days.

### 4. Self-Sabotaged OSS

Self-sabotage happens when legitimate maintainers decide to inject malicious content into their already popular OSS projects. There are many motivations for this type of attack, including hacktivism and revenge. In a recent incident that impacted thousands of projects, the author behind the two popular NPM packages, 'colors' and 'faker', intentionally committed bugs into the repositories and broke all the projects that depended on these packages. The motivation was the author's discontent with giant enterprises that use OSS without paying or contributing.

## OSS Dependency

In this section, we analyzed 70,000 source code repositories in production environments that Prisma Cloud monitors to understand the dependency relationship among OSS.

An application may depend on a list of OSS packages, and each OSS package may also depend on other OSS packages. As a result, all the directly or indirectly dependent packages can be unpacked into a tree that can go multiple levels deep. In a source code repository, we call the packages directly imported by the developer **root packages** and the packages imported by root packages **non-root packages**. When scanning an application for vulnerabilities, both root and non-root packages need to be considered. Our data shows that most of the packages in a repository are non-root packages and that non-root packages typically introduce most of the vulnerabilities.

Figure 11 shows the percentage of repositories categorized by the dependency count. Each slice represents a range of dependency counts. It is interesting to see that 31% of the repositories contain more than 400 dependent packages.

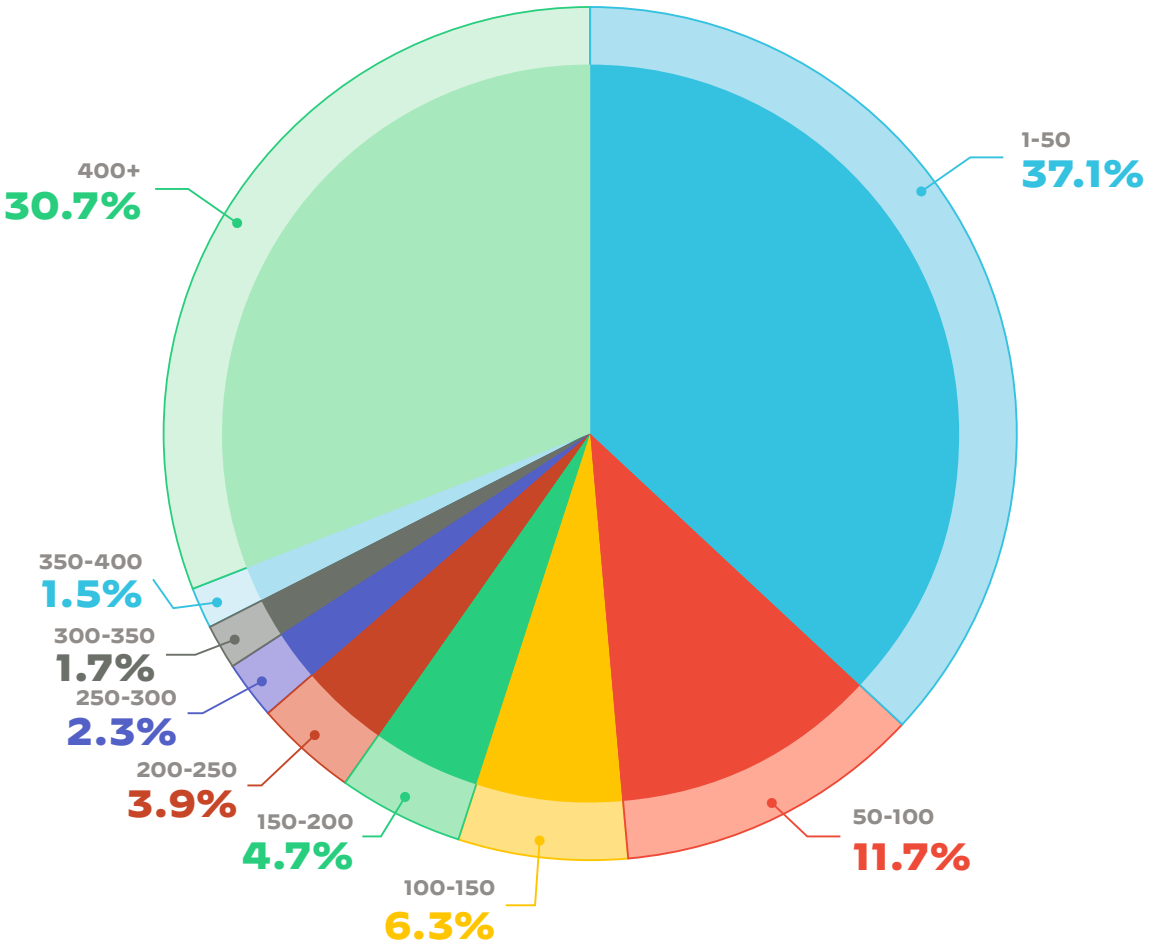



**Figure 11:** Percentage of repositories categorized by the dependency count

Figure 12 compares root packages and non-root packages. The left chart shows the percentages of root and non-root packages, and the right chart compares the percentages of vulnerabilities in the root and non-root packages. The numbers show a strong correlation between dependency count and vulnerability count:

- **51%** of codebases depend on more than **100** open-source packages.
- On average, each repository has **113** dependent OSS packages.
- **77%** of packages are non-root packages, and **77%** of vulnerabilities are introduced by the non-root packages.
- The number of vulnerabilities and security issues in a cloud workload is proportional to the number of external assets the workload depends on.

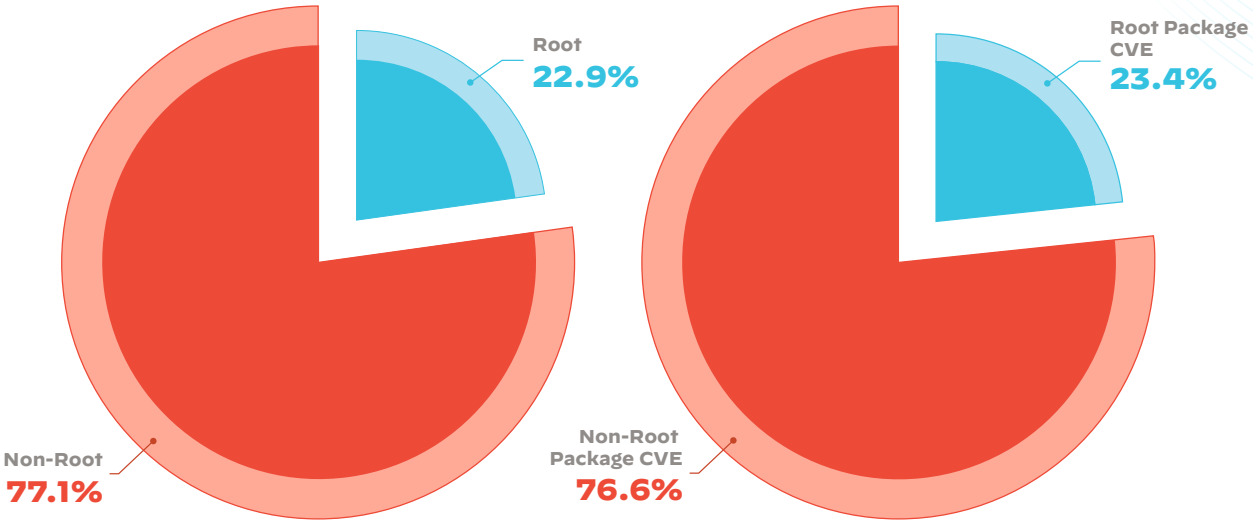



**Figure 12:** Comparison between root and non-root packages

OSS dependency does not stop at the package level. A container image can depend on other container images, and an IaC template (e.g., Helm, Terraform) can depend on other IaC templates. Consider an instance where each container contains 100 OSS packages and an IaC template imports 10 different container images. Then, deploying the IaC template will trigger thousands of OSS packages to be downloaded and executed. Any vulnerability or malicious artifact in these OSS packages may put the cloud infrastructure at risk.

Figure 13 shows the linear relationship between the number of dependent images and the number of vulnerabilities in container images. Images with more dependent images tend to have more vulnerabilities.

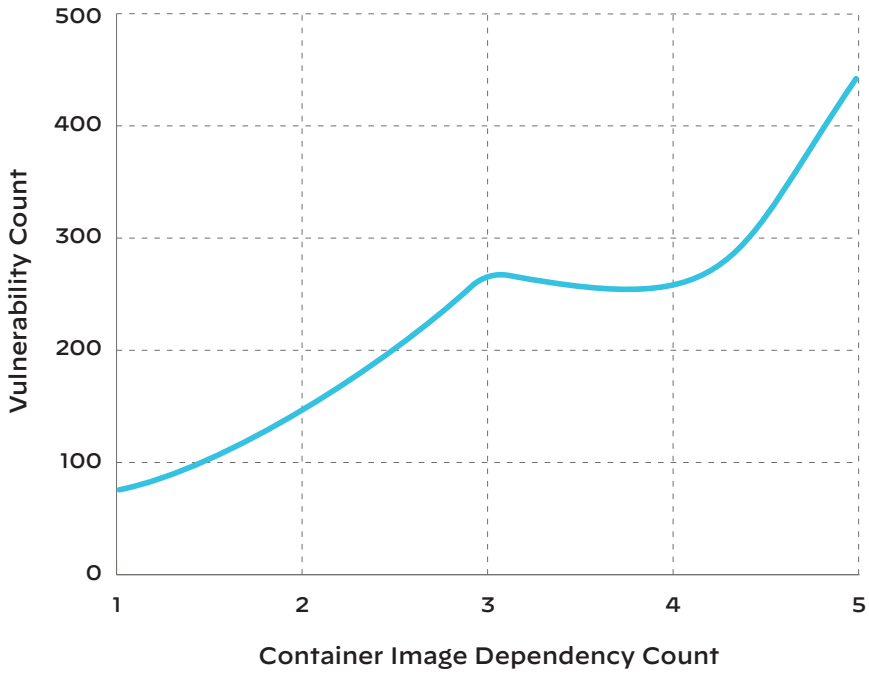



**Figure 13:** Number of the total vulnerabilities of a container image vs. the number of the dependent images of a container image

# Community and Government Responses

In light of the proliferation of OSS and the threats against OSS supply chain, the open-source community quickly responded with many projects focusing on identifying, accessing, and resolving security issues in OSS. The Go programming language community, for example, created Go Vulnerability Database and Govulncheck to help aggregate and normalize security issues that affect OSS written in Go. Another example is the Open Source Security Foundation (OpenSSF), which initiated the Alpha-Omega Project to identify and fix undiscovered vulnerabilities in open-source projects. 'Alpha' will work with the maintainers of the most critical open-source projects to identify and fix security vulnerabilities. 'Omega' will use automated methods and tools to identify critical security vulnerabilities across at least 10,000 widely deployed open-source projects.

Governments worldwide also see the criticality of OSS security.

The US government issued an executive order on improving the nation's cybersecurity. The order focuses on improving:

- Threat intelligence sharing between the government and the private sectors
- Software supply chain security
- Investigation and remediation capabilities

The executive order is also focused on modernizing and implementing security standards throughout the federal government.

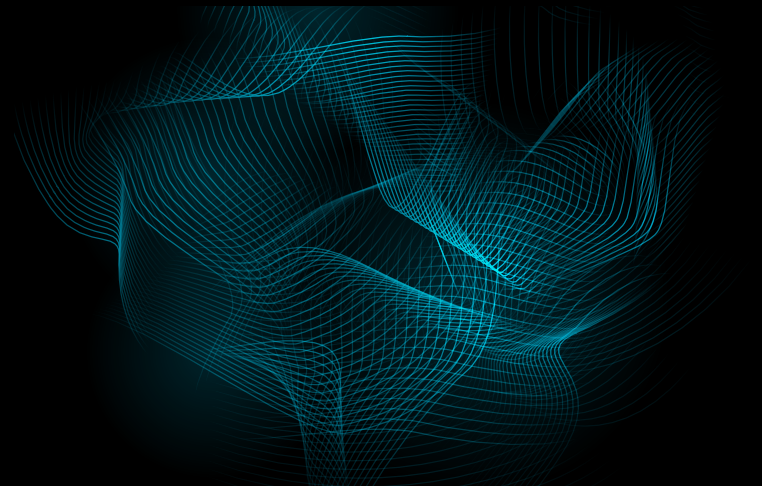
In early 2022, multiple government entities and private sector stakeholders convened at the White House to discuss how to improve the security of OSS and ways new collaboration could rapidly drive improvements.

The European Union initiated several new programs to counter the cyberthreat. In January 2022, the European Commission's Open Source Programme Office launched a bug bounty program to audit several open-source tools widely used by public services across the European Union. The European Commission is also currently legislating the Cyber Resilience Act, which will define new regulations for software or hardware products with digital elements. The act will bolster cybersecurity rules to ensure more secure hardware and software.

The German Federal Ministry for Economic Affairs and Climate Action launched the Sovereign Tech Fund to strengthen the digital infrastructure and open-source ecosystems in the public interest. Several popular open-source projects, including OpenMLS, Bundler/RubyGems, Sequoia-PGP, curl, WireGuard, and OpenSSH were granted the pilot round funding in late 2022.

It's encouraging to see governments and open-source communities across the globe respond with such urgency to software security issues. The success of these initiatives relies on the collaboration of governments, private sectors, and millions of developers. In the years to come, we look forward to more OSS regulation and support.

# Conclusion



This *Unit 42 Cloud Threat Report, Volume 7* is a comprehensive study of the cloud security landscape based on data collected from thousands of organizations in 2022. It explores a variety of security issues affecting the cloud, analyzes the details of actual attacks, and examines the impact of vulnerabilities in OSS.

What we learned throughout our research is that with cloud usage on the rise and cloud technologies continuing to mature, threat actors are getting smarter and more powerful every day, exploiting hidden weak spots and using vulnerabilities to their advantage. They've become masters at exploiting common issues like risky services and vulnerable applications exposed to the internet. They're developing TTPs that specifically target cloud workloads, and they're also utilizing cloud-native services to build command-and-control (C2) infrastructures and launch attacks. The wide adoption of OSS in the cloud drives risks even higher, making it faster and easier to compromise the shared software supply chain and ambush large numbers of victims simultaneously.

The bottom line to our findings is simple: your organization may not be as secure as you think. You're going to need to be vigilant, proactive, and innovative to stay ahead of adversaries.

Organizations should expect the attack surface of cloud-native applications to continue to grow as threat actors find increasingly creative ways to target the misconfiguration of cloud infrastructure, APIs, and the software supply chain itself. To guard against these threats, the industry will see a move away from point security solutions to CNAPPs that offer a full spectrum of capabilities across the application development lifecycle. This prediction is underscored by findings in the recent *State of Cloud-Native Security Report*[5] where 75% of survey respondents said the point security tools they use create blind spots (a further 80% said they'd benefit from a centralized security solution). Gartner echoes the assertion that there will be a significant uptick in CNAPP adoption, having reported a 70% jump in client inquiries regarding CNAPPS from 2021 to 2022.[6]

The only way to defend against the changing scope and severity of today's security threats is to always stay one step ahead of the attackers who are perpetrating them. Make it a priority to educate yourself on the latest threat vectors and implement robust security solutions that take a comprehensive platform approach to identify and eliminate threats in real time before they can compromise your environment.

5. *State of Cloud-Native Security Report 2023*, Palo Alto Networks, March 7, 2023. https://www.paloaltonetworks.com/resources/research/state-of-cloud-native-security-2023.
6. Neil MacDonald, Charlie Winckless, and Dale Koeppen, *Market Guide for Cloud-Native Protection Platforms*, Gartner, March 14, 2023. https://www.paloaltonetworks.com/resources/research/gartner-market-guide-cnapp.

# Methodology

Unit 42 researchers continuously monitored telemetry from multiple internal and public sources to understand the threats, attacks, and issues in cloud environments. The internal data primarily come from tens of thousands of sensors deployed in organizations across CSPs, industries, and countries. The public data come from sources such as NVD, GitHub, GitLab, and Docker Hub. This report analyzed the data collected between January 2022 and January 2023 from the following sources.

### Palo Alto Networks Prisma Cloud

Prisma Cloud is a cloud-native application protection platform (CNAPP) that protects code, infrastructure, workloads, data, and applications across multicloud and hybrid cloud environments.

### Palo Alto Networks Unit 42 Incident Response

Unit 42 incident response experts utilize cloud technologies, including Cortex XDR, Cortex Xpanse, and Prisma Cloud, to discover attack vectors, identify the extent of access and the data at risk, and take appropriate remediation actions.

### Palo Alto Networks Cortex Data Lake

Cortex Data Lake is scalable cloud-based storage that collects, integrates, and normalizes enterprises' security data combined with multiple sources of threat intelligence. Cortex Data Lake enables large-scale, AI-based analytics to identify and stop the most sophisticated attacks.

# Authors

**Jay Chen, Cloud Threat Report Lead Researcher**

Shaul Ben Hai, Sharon Ben Zeev, Mike Brewer, Daniel Haim Breger, Artur Oleyarsh, Nathaniel Quist, Aviv Sasson, Ariel Zelivansky

# Editors

Brian Barr, Emily Rodenhuis, Laura Novak, Samantha Stallings, Lysa Myers, Erica Naone

# Appendix

## Cloud Threat Actor TTPs

Within the following tables, you will find cloud-specific TTPs employed by each CTA group. For complete matrices with all the TTPs, please refer to the Unit 42 ATOM links.

**Legend:**

| Cloud-Specific TTPs | Cloud Credential Usage/ Discovery TTP | Container-Specific TTPs | Container Escape/ Resource-Specific TTPs |
|---|---|---|---|

**Table 1: PurpleUrchin TTPs (Unit 42 ATOM)**

| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Collection |
|---|---|---|---|---|---|
| T1133 - External Remote Services | T1609 - Container Administration Command | T1136.003 - Create Account | T1546 | T1612 - Build Image on Host | T1119 - Automated Collection |
| | T1610 - Deploy Container | T1546 - Event Triggered Execution | T1053.007 - Scheduled Task/Job | T1610 - Deploy Container | T1074.002 - Data Staged |
| | T1053.007 - Scheduled Task/Job | T1133 - External Remote Services | | T1578.002 - Modify Cloud Compute Infrastructure | |
| | | T1053.007 - Scheduled Task/Job | | | |

**Table 2: Kinsing TTPs (Unit 42 ATOM)**

| Initial Access | Execution | Persistence | Privilege Escalation | Defense Evasion | Credential Access | Collection |
|---|---|---|---|---|---|---|
| T1078 - Valid Accounts | T1610 - Deploy Container | T1078 - Valid Accounts | T1078 - Valid Accounts | T1078 - Valid Accounts | T1528 - Steal Application Access Token | T1613 - Container and Resource Discovery |
| T1078.003 - Valid Accounts: Local Accounts | | T1078.003 - Valid Accounts: Local Accounts | T1078.003 - Valid Accounts: Local Accounts | T1078.003 - Valid Accounts: Local Accounts | T1552 - Unsecured Credentials | |
| T1078.004 - Valid Accounts: Cloud Accounts | | T1078.004 - Valid Accounts: Cloud Accounts | T1078.004 - Valid Accounts: Cloud Accounts | T1078.004 - Valid Accounts: Cloud Accounts | T1552.001 - Unsecured Credentials: Credentials in Files | |
| | | | | T1610 - Deploy Container | T1552.004 - Unsecured Credentials: Private Keys | |

| Table 3: WatchDog TTPs (Unit 42 ATOM) | | | | |
|---|---|---|---|---|
| **Execution** | **Privilege Escalation** | **Defense Evasion** | **Credential Access** | **Discovery** |
| T1610 - Deploy Container | T1611 - Escape to Host | T1562 - Impair Defenses | T1528 - Steal Application Access Token | T1046 - Network Service Scanning |
| | | T1562.001 - Impair Defenses: Disable or Modify Tools | T1552 - Unsecured Credentials | T1518 - Software Discovery |
| | | T1562.003 - Impair Defenses: Impair Command History Logging | T1552.001 - Unsecured Credentials: Credentials in Files | T1518.001 - Software Discovery: Security Software Discovery |
| | | T1562.004 - Impair Defenses: Disable or Modify System Firewall | T1552.003 - Unsecured Credentials: Bash History | T1613 - Container and Resource Discovery |
| | | T1562.007 - Impair Defenses: Disable or Modify Cloud Firewall | T1552.004 - Unsecured Credentials: Private Keys | |
| | | T1562.008 - Impair Defenses: Disable Cloud Logs | T1552.005 - Unsecured Credentials: Cloud Instance Metadata API | |
| | | T1610 - Deploy Container | T1552.007 - Unsecured Credentials: Container API | |

| Table 4: 8220 TTPs (Unit 42 ATOM) | | | | | | |
|---|---|---|---|---|---|---|
| **Initial Access** | **Execution** | **Persistence** | **Privilege Escalation** | **Defense Evasion** | **Credential Access** | **Discovery** |
| T1078.003 - Valid Accounts: Local Accounts | T1610 - Deploy Container | T1078.003 - Valid Accounts: Local Accounts | T1078.003 - Valid Accounts: Local Accounts | T1078.003 - Valid Accounts: Local Accounts | T1110 - Brute Force | T1087.001 - Account Discovery: Local Account |
| | | T1136 - Create Account | | T1562.001 - Impair Defenses: Disable or Modify Tools | T1552 - Unsecured Credentials | T1087.004 - Account Discovery: Cloud Account |
| | | | | T1562.008 - Impair Defenses: Disable Cloud Logs | T1552.001 - Unsecured Credentials: Credentials in Files | T1518.001 - Software Discovery: Security Software Discovery |
| | | | | T1610 - Deploy Container | T1552.003 - Unsecured Credentials: Bash History | |
| | | | | | T1552.004 - Unsecured Credentials: Private Keys | |
| | | | | | T1552.005 - Unsecured Credentials: Cloud Instance Metadata API | |
| | | | | | T1552.007 - Unsecured Credentials: Container API | |

# About

## Prisma Cloud

Prisma® Cloud is a comprehensive cloud-native security platform with the industry's broadest security and compliance coverage—for applications, data, and the entire cloud-native technology stack—throughout the development lifecycle and across hybrid and multicloud deployments. Prisma Cloud's integrated approach enables security operations and DevOps teams to stay agile, collaborate effectively, and accelerate cloud-native application development and deployment securely.

Prisma Cloud analyzes more than 10 billion events every month. By proactively detecting security and compliance misconfigurations as well as triggering automated workflow responses, Prisma Cloud helps ensure you continuously and securely meet the demands of your dynamic cloud architectures. Prisma Cloud customers are actively alerted for vulnerabilities, insecure configurations, and potentially malicious activities mentioned in the report. To get started with Prisma Cloud, **request your free trial** today.

## Unit 42

Unit 42™ brings together our world-renowned threat researchers with an elite team of security consultants to create an intelligence-driven, response-ready organization. The Unit 42 Threat Intelligence team provides threat research that enables security teams to understand adversary intent and attribution while enhancing protections offered by our products and services to stop advanced attacks. As threats escalate, Unit 42 is available to advise customers on the latest risks, assess their readiness, and help them recover when the worst occurs. The **Unit 42 Security Consulting team** serves as a trusted partner with state-of-the-art cyber risk expertise and incident response capabilities, helping customers focus on their business before, during, and after a breach. The Cloud Security Assessment (CSA) offered by Unit 42 uses a threat-informed approach leveraging the Prisma Cloud platform to find the risks that matter, across all of your cloud environments.

paloalto® NETWORKS | UNIT 42™ BY PALO ALTO NETWORKS

3000 Tannery Way
Santa Clara, CA 95054

| | |
|---|---|
| Main | +1.408.753.4000 |
| Sales | +1.866.320.4788 |
| Support | +1.866.898.9087 |

www.paloaltonetworks.com