



**GitProtect**  
by Xopero ONE

REPORT

# The CISO's guide to DevOps threats

Global insights, event analysis,  
and security challenges unwrapped

by GitProtect Lab

---

# Contents

Introduction	3
Top DevOps threat facts to awaken curiosity	5
Issues heatmap	6
The State of GitHub Threat Landscape for GitHub Admins - 2024 In Review	7
The State of Atlassian Threat Landscape - 2024 In Review	49
The State of GitLab Threat Landscape - 2024 In Review	78
The State of Azure DevOps Threat Landscape - 2024 In Review	113
DevOps Backup Best practices	131
Conclusion	153

---

# Introduction

It's not a secret that DevOps has changed and uplifted the pace and structure of software development. It accelerated delivery cycles, enabled agile innovation, and reshaped workflows across industries. However, this velocity has come at a high cost - security.

**To understand which security issues were threatening GitHub, Azure DevOps, GitLab and Atlassian users in 2024, we dived into deep research, which resulted in this ebook.**

Well, in 2024 alone, GitHub, GitLab, Jira, Bitbucket, and Azure DevOps users faced a rising tide of cyber threats, outages, and security breaches that exposed the critical weaknesses of many DevOps pipelines.

Moreover, thousands of “*ghost accounts*” were used by cyber criminals on GitHub to distribute malware, while repositories and even comment sections became targets for malicious payloads like Lumma Stealer, BeaverTail, and NjRat. Although going through the GitHub Status, it's possible to notice that **GitHub's incident count decreased from 165 in 2023 to 124 in 2024**, it still resulted in around 800 hours of degraded service performance.

On the other hand, **the number of incidents on GitLab increased by 21% year-to-year. In 2024**, GitLab experienced 798+ hours of service performance degradation and 585+ hours of partial disruptions.

---

Another DevOps platform, Microsoft's Azure DevOps, also faced intense pressure with 111 incidents. They impacted services for 826 hours (around 103 working days), which is about 28% of your yearly working time if your working day is 8 hours.

What about Atlassian's Jira? A widely used tool for project and issue tracking, Jira, for example, saw a 44% year-over-year increase in reported incidents, rising from 75 in 2023 to 132 in 2024. All together, these incidents led to over 2131 hours of downtime - nearly 266 working days. While Bitbucket reported 38 incidents of different impact, totalling 200 hours of disruption.

These numbers are more than just data points - they're a clear warning. Thus, to ensure your data resilience, your organization should adopt DevSecOps, a model where security is embedded from the very beginning of the software development lifecycle, not bolted on as an afterthought.

# Top DevOps threat facts to awaken curiosity

Even a single vulnerability can be enough for attackers to open the “door” and disrupt your business-critical services... **GitLab patched a total of 153 vulnerabilities in 2024.**

**It takes a seasoned developer about an hour and 21 minutes to set up a basic CI/CD pipeline** - the exact amount of time Azure DevOps was down across North and South America.

**Azure DevOps faced around 826 hours of disruptions in 2024.** What can a developer do during this time? Compete in 8 to 10 hackathons, for example.

**In 2024, GitLab experienced over 798 hours of service disruptions.** You can backpack Europe, explore Southeast Asia, or cross the US coast-to-coast during this time.

**Bitbucket experienced critical and major disruptions for over 70 hours,** which equals the time you need to binge-watch the entire Game of Thrones series from start to finish.

**GitHub suffered 26 major incidents during 2024,** which resulted in over 134 hours of disruptions, **which is more than 3 working weeks.**

**In Q3 alone, Jira users faced over 7 hours of critical disruptions** - it's nearly a full workday. During this time, a Jira Admin could run an entire Agile sprint planning session, from backlog grooming to story point estimation and stakeholder alignment.

**Atlassian's Confluence experienced 1,963 hours of disruptions in 2024,** which is equivalent to nearly 67% of a full working year.

# Issues heatmap

**955 hrs of disruptions in total**, which is enough to cross the Atlantic by small yacht, with a short break in the Caribbean, reach the East Coast and then go back to Europe.






**GitHub:** 26 major impact issues / 134hrs of disruption

**BitBucket:** 4 major impact issues / 4hrs of disruption

**Jira:** 10 major impact issues / 17hrs of disruption

**GitLab:** 7 major impact issues / 798hrs of disruption

**Azure DevOps:** 1 major impact issue / ~2hrs of disruption

	 GitHub	 Bitbucket	 Jira	 GitLab	 Azure DevOps
January	9	2	7	4	no data
February	9	10	15	6	12
March	8	2	7	6	7
April	18	2	5	10	12
May	11	2	10	3	10
June	8	6	6	5	15
July	20	5	5	12	22
August	11	2	15	7	5
September	11	4	18	17	6
October	4	0	19	9	7
November	7	2	14	8	12
December	8	1	11	10	3


## Total number of issues publicly announced:

**GitHub:** Minor Impact, Major Impact, No Impact, Maintenance

**Bitbucket & Jira:** Minor Impact, Critical Impact, No Impact, Major Impact, Maintenance

**GitLab:** Partial Service Disruption, Degraded Performance, Operational, Service Disruption, Planned Maintenance, Security Issues

**Azure DevOps:** Unhealthy, Degraded

 GitHub 2024 report

# The State of GitHub Threat Landscape for GitHub Admins - 2024 In Review

26

major impact issues

134<sub>hrs</sub>

of disruption



GitProtect  
by Xopero ONE

# The State of GitHub Threat Landscape for GitHub Admins – 2024 In Review

**In 2024 threat actors were trying to abuse GitHub even more severely than before.** There were detected thousands of “ghost accounts” spreading malware. Hackers were also trying to succeed in their malware campaigns using not only repositories but also comments.

Moreover, multiple organizations that store their source code on GitHub have experienced data leakage and data breaches. Among those companies are The New York Times, Mercedes Benz, Ultralitics, etc.

However, when it comes to the incidents that GitHub faced in 2024, its number reduced by about 25% with the total number of reported issues on GitHub Status of 124. **Last year, the service provider experienced 165 incidents of different severity.**

## GitHub Status info: 8 issues

DECEMBER '24

Incidents that had a major impact – 2 incidents

**Total time of partial disruption – 59 minutes**

Incidents that had a minor impact – 6 incidents

**Total time of degraded performance – 15 hours 7 minutes**

In December, GitHub experienced 2 major incidents, causing around an hour of partial disruption. Additionally, there were 6 minor incidents that resulted in over 15 hours of degraded performance, primarily affecting GitHub Enterprise Cloud in the EU, Pull Requests, and Actions services.

## GitHub recorded incidents during the year

**124**

incidents total

■ 2024  
■ 2023



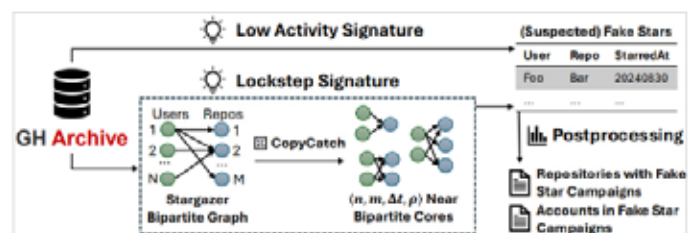
GitProtect.io

# 3M+ fake “star” GitHub projects used to spread malware

GitHub is facing an issue with inauthentic “stars” used to inflate the popularity of scam and malware-distributing repositories, making them appear trustworthy to users.

A study by researchers from Socket, Carnegie Mellon University, and North Carolina State University State identified 4.5M+ suspected fake stars across 22,915 repositories, with many stemming from coordinated bot-like activity.

Using the tool StarScout, the team analyzed GitHub metadata and flagged repositories with suspicious patterns, reducing the list to 3.1 million fake stars involving 15,835 repositories. Approximately 91% of flagged repositories were deleted by October 2024, but fake star activity continues to rise, with significant spikes in 2024.



These fraudulent practices erode trust in GitHub and its projects, emphasizing the need for users to evaluate repositories carefully beyond their star count. GitHub is actively removing flagged repositories and accounts but remains challenged by ongoing malicious campaigns.

[Bleeping Computer](#) / [SC Media](#)

---

# Software developers are targeted by the North Korean hackers

The North Korean “*Contagious Interview*” hacker campaign targets aimed at developers with fake job postings to distribute malware like BeaverTail and OtterCookie.

OtterCookie infects systems via loaders in Node.js projects, npm packages, or applications downloaded from any of GitHub or Bitbucket, later using Socket.IO WebSocket for command-and-control communications.

The malware’s functionalities include stealing crypto wallet keys, sensitive clipboard data, and performing reconnaissance with shell commands. This campaign highlights the attackers’ evolving techniques and emphasizes the need for developers to verify job-related information and scrutinize code from untrusted sources.

[Techzine](#) / [SC Media](#)

---

## NjRat 2.3D Pro Edition might be actively exploited on GitHub

The discovery of NjRat 2.3D Professional Edition on GitHub raises significant concerns because of its advanced capabilities and ease of access to cybercriminals.

NjRat, a Remote Access Trojan active since 2012, enables attackers to steal credentials, log keystrokes, access webcams, and deploy additional malware. The latest version, which is publicly available on GitHub, introduces enhanced features for remote control and data theft. It even more increases its threat level. Open access on such platforms lowers barriers for threat actors, potentially leading to a surge in cyberattacks.

Thus, to eliminate the threat organizations must adopt [robust cybersecurity measures](#), including endpoint protection, employee awareness, regular updates, and network monitoring, while platforms like GitHub need to improve their mechanisms to prevent hosting malicious software.

[Trading View](#)

# Major GitHub outage leaves thousands of developers locked out of their projects

On December 17th, more than 1,3K users in UK and 2K in the United States reported issues in GitHub services work, according to Downdetector.

17TH DEC 2024, 15:29

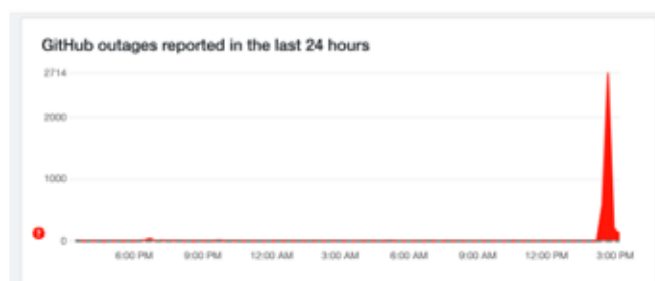
By Allan Glen

## Reports begin to drop

At its peak, nearly 3,000 users in the US were reporting issues with the software although this has recently dropped to several hundred.

This would indicate the issue is close to being resolved.

Thanks for following us. We will keep you updated in this blog should any further developments occur.



As GitHub later explained in its Status, *“the increased error rate caused a broad impact across our services, such as the inability to log in, view a repository, open a pull request, and comment on issues”*.

The reason of why *“many users experienced intermittent errors and timeouts”* when were trying to access GitHub took place due to planned maintenance that caused the live updates service to fail and overloaded web servers.

[The Sun](#)

---

# Hacker stole over 390K WordPress credentials through their malicious GitHub repo hosting PoC exploits

A malicious GitHub repository, now removed, enabled the exfiltration of 390K+ credentials, primarily targeting WordPress accounts, through a fake tool called “*Yet Another WordPress Poster*” (yawpp).

The repository, associated with a threat actor dubbed MUT-1244, also deployed malware via a rogue npm dependency and phishing emails. Victims included pentesters, security researchers, and malicious actors who inadvertently exposed sensitive data such as SSH private keys and AWS credentials. MUT-1244’s tactics include creating trojanized GitHub repositories hosting fake PoC exploit code and employing phishing emails to deliver payloads like cryptocurrency miners and data theft tools.

Such campaigns often exploit vulnerability disclosures, using fake GitHub profiles with AI-generated images to lure victims. The second-stage malware is distributed through multiple methods, including backdoored files, malicious PDFs, Python droppers, and npm packages. The attacks highlight the growing risks of compromised PoC repositories and phishing targeting the cybersecurity community.

[The Hacker News](#) / [SC Media](#) / [HACKread](#)

---

## A compromised Ultralytics AI library – urgent update needed

There were found 2 compromised versions of the Python AI library Ultralytics (8.3.41 and 8.3.42) to contain a cryptocurrency miner, exploiting the build environment through a GitHub Actions script injection vulnerability.

The attack caused discrepancies between the source code on PyPI and the GitHub repository, with malicious pull requests originating from an account linked to the OpenIM SDK. The miner, identified as XMRig, triggered high CPU usage on macOS and Linux systems, but experts warn the attack could have been more severe if backdoors or RATs were planted.



*"It seems that the malicious payload served was simply an XMRig miner, and that the malicious functionality was aimed at cryptocurrency mining..[...] But it is not hard to imagine what the potential impact and the damage could be if threat actors decided to plant more aggressive malware like backdoors or remote access trojans (RATs)," says a ReversingLabs engineer.*

The compromised versions have been removed from PyPI, and a secure update has been released to fix the issue, with users strongly advised to upgrade immediately.

[The Hacker News](#)

## GitHub Status info: 7 incidents

NOVEMBER '24

Incidents that had a major impact – 1 incident

**Total time of partial disruption – 1 hour 7 minutes**

Incidents that had a minor impact – 6 incidents

**Total time of degraded performance – 12 hours 10 minutes**

In November 2024, GitHub users might notice 7 incidents that the service provider experienced.

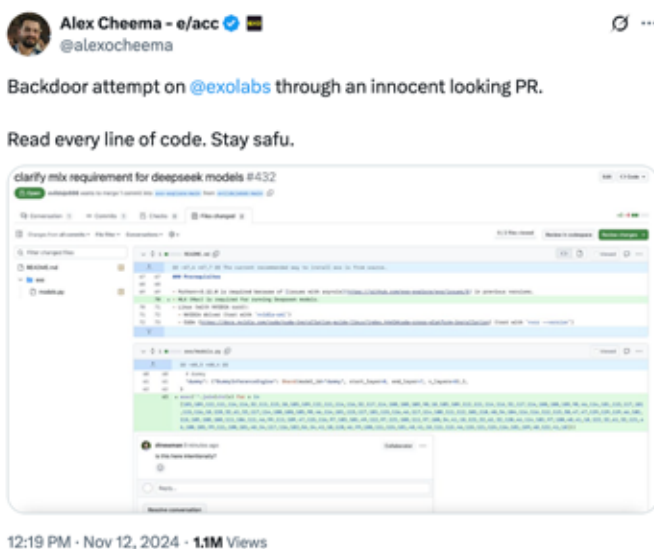
There was a major incident on November 19 lasted over an hour, delaying notifications due to a database issue and 6 minor incidents, totaling 12 hours and 10 minutes of degraded performance, included issues with Codespaces, GitHub Search, Copilot Chat, and merge queues.

# Multiple GitHub projects are targeted with malicious commits

A number of GitHub projects, including EXO Labs, were targeted by malicious commits attempting to inject backdoors into repositories. One of such pull request disguised as an innocent code change used Unicode obfuscation to introduce a script designed to fetch and execute remote code, though the associated URL contained no payload.

The roots of the attack goes to accounts impersonating Texas-based security researcher and ethical hacker Mike Bell, who denied involvement, claiming it was a smear campaign.

Other open-source projects, like “yt-dlp,” faced similar attacks, with at least 18 malicious pull requests that were identified.



After this threat was identified, many impersonating accounts and associated commits were removed by GitHub. Tools like Presubmit’s AI Reviewer successfully flagged these threats during testing, emphasizing the importance of automated and human code reviews. This incident proves ongoing risks in open-source supply chains and the need for vigilant monitoring of all the contributions.

[Bleeping Computer](#)

---

# Hackers can execute malicious commands due to GitHub CLI RCE Vulnerability

A critical vulnerability in GitHub CLI, tracked as CVE-2024-32002 might allow an attacker to execute arbitrary code on a user's system via remote code execution when interacting with malicious Codespace environments. The flaw, which was found in versions prior to 2.62.0, exploits how SSH connection details, including crafted usernames, are handled in commands like `gh codespace ssh`. Attackers can inject malicious SSH arguments through dev containers, potentially compromising sensitive data, installing malware, or enabling further network intrusion.

To mitigate the potential threats, users are urged to update to version 2.62.0 or later, use trusted dev containers, and avoid connecting to untrusted Codespaces to mitigate this risk.

[Cyber Security News](#)

---

# GitHub developers are targeted by Golssue

The GitLoker hacking group that has been hijacking GitHub repos for a while, introduced a new phishing tool, Golssue, the aim of which is to extract GitHub user email addresses and automate phishing campaigns targeting developers and their organizations.

This tool enables large-scale credential theft, supply chain attacks, and source code breaches by using features like customizable email templates, email scraping, and token management.

Cybersecurity experts warn that Goissue's automation and scalability might significantly increase the threat level, exploiting trust within the developer community to hijack repositories and compromise networks.

[The Hacker News](#) / [Dark Reading](#) / [Security Week](#)

## **GitHub Status info: 4 incidents**

OCTOBER '24

Incidents that had a major impact – 1 incident

## **Total time of partial disruption – 7 hours 18 minutes**

Incidents that had a minor impact – 3 incidents

## **Total time of degraded performance – 11 hours 10 minutes**

---

GitHub users might experience 4 incidents of different importance in October. The most notable of them lasted for over 7 hours and involved DNS infrastructure failures after a database migration, affecting such services as Copilot, Actions, customer migrations, support, and code search.

---

## Cisco's DevHub's environment is hacked

Cisco confirmed that some files were stolen after a hacker, IntelBroker, claimed a breach involving GitHub and SOnarQube project, source code, credentials, confidential documents, certificates, API tokens, Jira tickets, encryption keys, AWS private buckets, and other data.

Though, Cisco's investigation found no breach of its systems but revealed that the hacker exploited a public-facing DevHub environment used for customer resources. The company determined that only a small number of files were exposed, with no evidence so far of sensitive PII or financial data being included.

Public access to the affected website has since been disabled as the investigation continues. IntelBroker is known for targeting major organizations, though victims, including Deloitte, Microsoft, AT&T, Verizon, etc. often report limited impact from such incidents.

[Security Week](#)

---

# GitHub's security flaw might allow an attacker to access an instance without authorization

In October GitHub released security updates for Enterprise Server (GHES) to address multiple vulnerabilities, including a critical issue, tracked as CVE-2024-9487 with the CVSS score of 9.5 out of the maximum of 10.

The vulnerability could allow attackers to bypass SAML SSO authentication and gain unauthorized access. This flaw was a regression from a previous patch for another severe vulnerability, CVE-2024-4985 with a severity of 10.0

Two additional issues were also fixed: an information disclosure vulnerability, tracked as CVE-2024-9539 with a CVSS score of 5.7 related to SVG metadata and sensitive data exposure in HTML forms within the management console. These vulnerabilities have been resolved in GHES versions 3.14.2, 3.13.5, 3.12.10, and 3.11.16. GitHub strongly advises organizations to update to the latest versions to mitigate security risks.

---

# GitHub links are exploited in a new malware campaign targeted at the finance and insurance industries

According to the cybersecurity firm, Confense, cybercriminals created a new scheme to target finance and insurance companies. In this cyberattack campaign, they sent victims tax-themed phishing emails with trusted GitHub links to deliver the Remcos Remote Access Trojan (RAT), bypassing security measures. Attackers exploit legitimate repositories like UsTaxes and HMRC, uploading malicious files as comments and deleting them while keeping the malware link active. All of that makes the detection challenging.



*“In this campaign, legitimate repositories such as the open-source tax filing software, UsTaxes, HMRC, and InlandRevenue were used instead of unknown, low-star repositories. Using trusted repositories to deliver malware is relatively new compared to threat actors creating their own malicious GitHub repositories. These malicious GitHub links can be associated with any repository that allows comments,” states [Cofenser](#) in its research.*

This tactic, effective due to GitHub’s reputation as a trusted domain, represents a shift from traditional methods where hackers relied on their own repositories.

It only proves the fact that as the attacks grow more sophisticated, meaning that businesses in targeted industries must strengthen cybersecurity measures to safeguard against evolving threats.

[The Hacker News](#) / [Entrepreneur](#) / [Dark Reading](#)

### GitHub Status info: 11 incidents

SEPTEMBER '24

Incidents that had a major impact – 3 incidents

**Total time of partial disruption – 2 hours 36 minutes**

Incidents that had a minor impact – 7 incidents

**Total time of degraded performance – 230 hours 30 minutes**

Operational issues – 1 incident

**Total time of disruption – 8 minutes**

In September 2024, GitHub experienced 11 incidents, which included 3 major disruptions totaling over 2 hours and 7 minor incidents causing 230+ hours of degraded performance.

The issues included Codespaces interruptions in Central India and network connectivity failures affecting Codespaces, degraded Actions and Pages services, Copilot degradation, and widespread delays in Issue, Pull Requests, and Webhook processing.

---

# Attackers abuse GitHub repos to distribute the password-stealing malware

The threat actors built their phishing campaign by exploiting GitHub's "Issues" feature. Thus, they managed to distribute the Lumma Stealer malware by tricking users into visiting a fake "*GitHub Scanner*" website.

How does it work? Threat actors open issues on open-source repositories, falsely claiming security vulnerabilities and directing users to counterfeit sites. The campaign's credibility is enhanced by legitimate GitHub email notifications sent to repository contributors, making the alerts appear authentic. Upon visiting the fraudulent site, users encounter a fake CAPTCHA that covertly executes malicious JavaScript, which downloads and installs the Lumma Stealer malware. This malware steals credentials, authentication cookies, browsing history, cryptocurrency wallets, and sensitive files.

The attackers aim to compromise developers' credentials, potentially modifying source code to execute supply chain attacks.

Thus, to mitigate the risks, users are advised to avoid interacting with suspicious links and report such issues to GitHub immediately.

[Bleeping Computer](#) / [Gigazine](#)

---

## Around 3K of GitHub accounts might be spreading a malware

Experts from [Check Point Research](#) found out that cybercriminals are leveraging ghost accounts on GitHub to distribute malware through a sophisticated distribution-as-a-service (DaaS) network called the Stargazers Ghost Network.

These accounts show normal activities to avoid detection while distributing malicious links via platforms like Discord, targeting users interested in boosting social media followers.

The network has hosted over 2,200 malicious repositories, spreading malware like Atlantida Stealer, which steals credentials, cryptocurrency wallets, and personal information, infecting over 1,300 victims in just 4 days during a campaign. The scheme also generates income through

dark web services like selling stars, followers, and trusted accounts, earning approximately \$100,000.

GitHub has taken steps to disable these accounts, but researchers believe this is only the beginning, as similar operations likely span other platforms like YouTube, Instagram, and Facebook. Future campaigns may use AI-driven accounts to generate targeted, personalized phishing material, making malware distribution harder to detect. Check Point Research warns this marks a new era in cyber threats requiring heightened vigilance.

[Security Intelligence](#) / [The Register](#)

## GitHub Status info: 11 incidents

AUGUST '24

Incidents that had a major impact – 1 incident

**Total time of partial disruption – 36 minutes**

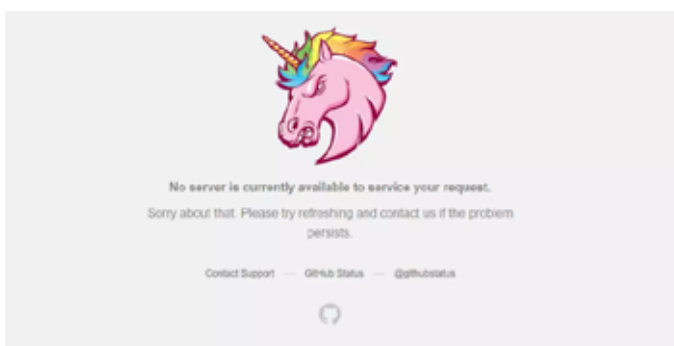
Incidents that had a minor impact – 10 incidents

**Total time of degraded performance – 12 hours 2 minutes**

In August 2024, GitHub experienced 11 incidents, including 1 major disruption where all GitHub services were inaccessible for more than 30 minutes. Between 23:02 UTC and 23:38 UTC on August 14th, GitHub users all over the world couldn't access the service and were shown a picture of an angry pink unicorn while the service provider was downtime.

According to [GitHub Status](#), the incident happened “*due to a configuration change that impacted traffic routing within our database infrastructure, resulting in critical services unexpectedly losing database connectivity. There was no data loss or corruption during this incident.*”

The remaining 10 minor incidents that GitHub saw in August caused 12+ hours of degraded performance in Actions, Copilot outages that affected 25% of users, and elevated API error rates.



Source: [Times Now World](#)

---

## Hackers use GitHub comments to push malware

Another case of Lumma Stealer was reported by Bleeping Computer in late August. This time it was found that threat actors were abusing GitHub by posting fixes in project comments, distributing a password stealing malware, which steals credentials, cookies, cryptocurrency wallets, and sensitive files.

The bad actors have posted thousands of malicious comments, often containing links to password-protected archives with executables across GitHub projects. All of that is to trick users into downloading and running the malware. Over 29K of such comments were posted in just 3 days.

Though, GitHub immediately searched and deleted them, some of the users were fallen victim. The service provider advises the affected users to reset their passwords and migrate cryptocurrencies to secure wallets.

[Bleeping Computer](#)

---

# GitHub security flaw ArtiPACKED can allow a takeover of GitHub repos

Attackers can exploit ArtiPACKED, a detected attack vector in GitHub Actions artifacts, to take over GitHub repositories and also access the organization's cloud environments.

The given vulnerability exploits misconfigurations and security flaws to expose sensitive tokens and compromise repositories and cloud environments. With its help threat actors can exploit leaked GitHub tokens, such as `GITHUB_TOKEN` and `ACTIONS_RUNTIME_TOKEN`, to gain unauthorized access, inject malicious code, and execute remote code. These artifacts, which store workflow data like builds and logs, are publicly accessible in open-source projects, making them a high-risk target for attackers. The issue is exacerbated by race conditions and improperly secured workflows, enabling attackers to replace artifacts or steal tokens during active CI/CD pipeline runs.

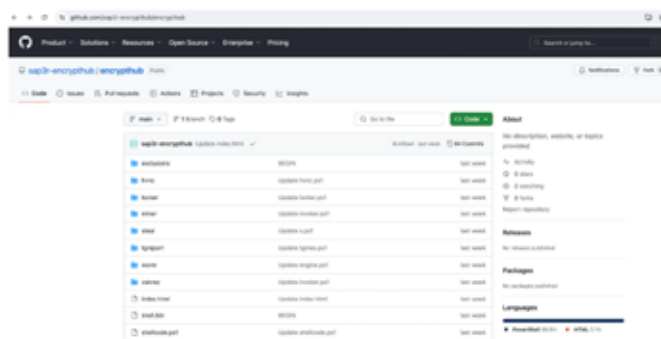
GitHub has classified the issue as informational, urging users to secure their artifacts and reevaluate usage, especially following the deprecation of Artifacts V3.

[The Hacker News](#) / [Bleeping Computer](#) / [SC Media](#)

# Fake WinRar website delivers ransomware via GitHub

The cyber security researchers at Sonic Wall found out that threat actors can exploit fake websites, like a deceptive WinRar lookalike (win-rar[.]co), to distribute malware and steal personal data.

This typosquatting site hosts a malicious shell script, zx.ps1, which downloads additional malware components from a GitHub repository named “*encrypthub*.” The repository contains tools for ransomware, crypto mining, information theft, and shellcode injection, among other malicious capabilities, with components sending system data to a Telegram account.



Source: Cybersecurity news  
(Main malware project page hosted on GitHub)

While there haven't been proven full-scale attacks using all components, its infrastructure reveals the potential for complex, multi-stage intrusions. Users are urged to verify software sources and exercise caution when downloading programs to mitigate such threats.

[Cyber Security News](#)

It only proves the fact that as the attacks grow more sophisticated, meaning that businesses in targeted industries must strengthen cybersecurity measures to safeguard against evolving threats.

### GitHub Status info: 20 incidents

JULY '24

Incidents that had a major impact – 3 incidents

**Total time of partial disruption – 63 hours 58 minutes**

Incidents that had a minor impact – 17 incidents

In July 2024, GitHub experienced 20 incidents, including 3 major disruptions totaling 63 hours and 58 minutes of partial service outages. The most significant major incidents included a 19-hour Copilot outage, a Copilot Chat disruption, and delayed webhook deliveries. Among the 17 minor incidents (totaling over 50 hours of degraded performance), notable issues included regional Codespaces and Actions delays, Copilot Chat timeouts, and GitHub documentation outages. A prolonged 29-hour degradation affected external identity linking, highlighting the impact of infrastructure and provider-related issues. These disruptions underscore the challenges of maintaining seamless service for a globally distributed platform.

---

# Python's core repos exposed by GitHub token leak

Cybersecurity researchers at JFrog detected a leaked GitHub token in a public Docker container on Docker Hub that could have granted elevated access to critical Python repositories, including PyPI and the Python Software Foundation.

If exploited, this token could have enabled a large-scale supply chain attack by injecting malicious code into Python packages or even the Python language itself. The token, tied to PyPI Admin Ee Durbin, was immediately revoked after responsible disclosure on June 28, 2024, with no evidence of exploitation in the wild.

The leak occurred due to a developer's local misconfiguration while bypassing GitHub API rate limits, unintentionally leaving the token in the code.

Separately, Checkmarx identified malicious PyPI packages linked to a Telegram bot that exfiltrates sensitive data, showing the persistent risks of supply chain and information theft attacks.

[The Hacker News](#)

## GitHub Status info: 8 issues

JULY '24

Incidents that had a major impact – 2 incidents

**Total time of partial disruption – 3 hours 20 minutes**

Incidents that had a minor impact – 6 incidents

**Total time of degraded performance - 26 hours 34 min**

In June 2024, GitHub reported in its Status 8 incidents, including 2 major disruptions lasting a total of 3 hours and 20 minutes. These involved a migration service outage caused by invalid credentials and a degraded Issues service that hid project events on timelines.

Other 6 minor incidents, totaling 26 hours and 34 minutes, included delays in organization membership updates, regional Codespaces issues, and a prolonged Copilot Pull Request Summaries outage. Other minor disruptions affected migrations, Actions, and the Maven package registry.

# A 270GB New York Times source code leaked

According to the message of the 4chan user, 270GB of internal New York Times data, including alleged source code for Wordle, internal communications, and sensitive authentication details from 5K GitHub repositories.

but stated no unauthorized access to their systems or operational impacts occurred. They emphasized that appropriate measures were taken at the time and that their security includes ongoing monitoring for anomalies.



Source: [Bleeping Computer](#)

According to the readme file in the archive, the attacker managed to exploit the exposed GitHub token to access the New York Times' GitHub repositories and steal the data.

The New York Times confirmed a January 2024 incident where credentials to a third-party code platform were inadvertently exposed



*"The underlying event related to the recent online posting of Times information occurred in January 2024, when a credential to a cloud-based third-party code platform was inadvertently made available," says Charlie Stadtlander, New York Times managing director for external communications, newsroom, and opinion. "The issue was quickly identified, and we took appropriate measures in response at the time. There is no indication of unauthorized access to Times-owned systems nor impact to our operations related to this event. Our security measures include continuous monitoring for anomalous activity." – The New York Times*

[Information Week](#)  
[Bleeping Computer](#)

---

# GitHub, npm, and jsDelivr code repos compromised with trojanized jQuery packages

Threat actors have been distributing trojanized versions of jQuery through npm, GitHub, and jsDelivr in a complex and persistent supply chain attack targeting website form data. The attack involved 68 manually crafted packages published over nearly a month, with malicious code hidden in the seldom-used “*end*” function of jQuery, which is called by the popular “*fadeTo*” function. These trojanized packages exfiltrate data to a remote URL and are hosted on a GitHub repository linked to the account “*indexsc*,” with additional scripts using jsDelivr to appear legitimate and evade firewalls.

This attack highlights a growing sophistication in supply chain threats, with tailored manual efforts replacing automated tactics to exploit widely trusted platforms.

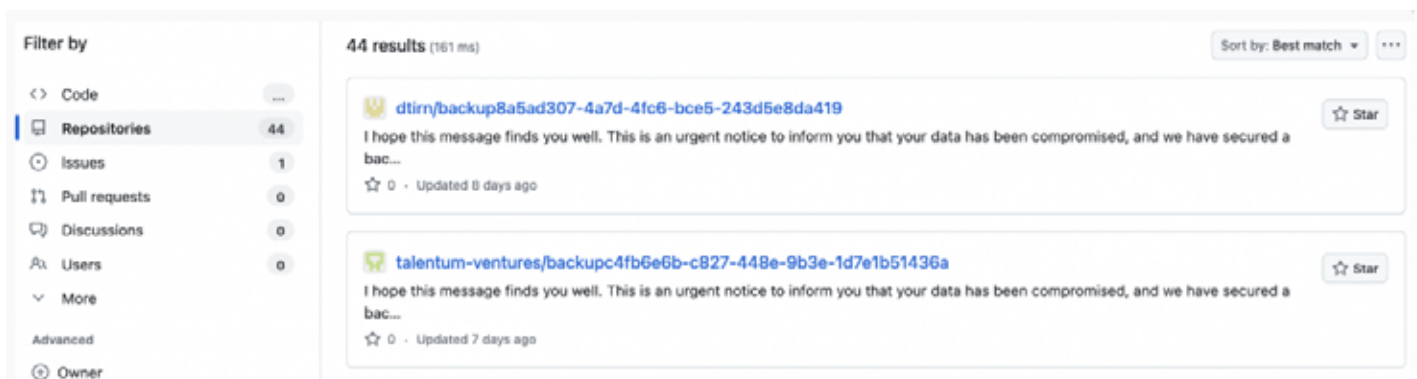
[The Hacker News](#)

# GitHub repositories wiped in Gitlocker ransomware attacks

In their attacks, Gitlocker, a threat actor, targets GitHub repositories and wipes their content to ask for a ransom.

In their malicious schemes, a threat actor steals the victim's GitHub data, creates a backup and deletes the user's data. Then, they rename the repo, add a README.me file and reach out to the victim on Telegram to demand a ransom.

*"I hope this message finds you well. This is an urgent notice to inform you that your data has been compromised, and we have secured a backup,"* the ransom message may state.



Source: [Bleeping Computer](#)

According to Bleeping Computer, dozens of GitHub repositories were compromised. GitHub advances its users to follow security best practices, including rotating their passwords, adding new SSH keys, modifying team members, having a backup, etc.

## GitHub Status info: 11 issues

MAY '24

Incidents that had a major impact – 11 incidents

**Total time of partial disruption – 41 hours 48 minutes**

GitHub experienced 11 incidents in May 2024. The total time of degraded performance compounded over 40 hours. Issues ranged from service disruptions affecting Copilot, Codespaces, and Actions to prolonged outages rendering Jupyter Notebooks and creating VMs in Azure.

---

# Critical GHES vulnerability can allow an attacker to bypass authentication

GitHub patched a high-severity vulnerability, tracked as CVE-2024-4985 with the highest possible CVSS score of 10, which might allow a threat actor to bypass the authentication protection. The vulnerability was detected in GitHub Enterprise Server versions prior to 3.13.0.

To mitigate the risks, GitHub recommends the organizations that use the affected versions update their instances to the latest version of GHES.

[The Hacker News](#) / [The Register](#)



*“On instances that use SAML single sign-on (SSO) authentication with the optional encrypted assertions feature, an attacker could forge a SAML response to provision and/or gain access to a user with administrator privileges,” GitHub communicated in its [advisory](#).*

---

## GitHub data breach affects fintech company

Iress, a financial services software firm, confirmed in May that there was an unauthorized access to its GitHub user space that resulted in stolen credentials being used to access its OneVue production environment, which contains client data.

While initially stating no client data was compromised, the firm now acknowledges the potential data exposure and is investigating the extent and nature of the breach.

Iress has stressed that the issue is isolated to the OneVue business and that no evidence suggests other parts of its production environment or client software were affected. The company has implemented stronger security protocols and emphasized that there is no anticipated disruption to its operations or client services.

[Cyberdaily](#)



*“Investigations have substantially progressed across Iress’ other business lines, and at this time, we have found no evidence that the remainder of Iress’ production environment, software or client data has otherwise been compromised,” the fintech company stated.*

# Over \$300M was stolen from the crypto exchange DMM

North Korean-linked hackers stole 4,502.9 Bitcoin (at the time of the hack it completed around \$305M!) from the Japanese crypto exchange DMM, leveraging a sophisticated social engineering attack.

In the mutual investigation of the FBI and NPA of Japan, they found out that the TraderTraitor group impersonated a recruiter on LinkedIn, targeting an employee at Ginfo, a crypto wallet company. The hacker tricked the employee into running malicious code disguised as a pre-employment test, compromising their GitHub account and gaining access to sensitive systems.



In May, the group exploited this access to manipulate a legitimate transaction request, transferring 4,502.9 Bitcoin to wallets they controlled. The FBI and its international partners continue efforts to expose and counter North Korea's cyber operations to fund its regime.

## GitHub Status info: 18 issues

APRIL '24

Incidents that had a major impact – 5 incidents

**Total time of partial disruption – 77 hours 59 minutes**

Incidents that had a minor impact – 13 incidents

**Total time of degraded performance – 132 hours 23 minutes**

In April 2024, GitHub reported 18 issues, including 5 major incidents totaling around 78 hours of partial disruptions, and 13 minor incidents with over 132 hours of degraded performance.

The incidents included delays in pull request operations, degraded Codespaces creation, and intermittent failures in GitHub Actions and NPM package downloads.

# Acuity's GitHub repos were breached leaking non-sensitive govt data

A tech consulting firm and a federal contractor affiliated with the US government agencies, Acuity, confirmed a breach of its GitHub repositories, involving the theft of outdated and non-sensitive data.

The breach was allegedly carried out by threat actors, including IntelBroker, who exploited a vulnerability in Acuity's Tekton CI/CD server to steal credentials and access private repositories.



*"After conducting our own analysis and following a third-party cybersecurity expert investigation, Acuity has seen no evidence of impact on any of our clients' sensitive data. In addition to cooperating with law enforcement, Acuity takes the security of its customers' data seriously and is implementing appropriate measures to secure its operations further."*

*Source: Bleeping Computer – Acuity's official statement*

While Acuity stated that no sensitive client data was impacted, IntelBroker has leaked records purportedly containing information related to State Department, Justice Department, DHS, and FBI employees. Moreover, the hacker claims to have stolen classified Five Eyes intelligence documents.

The breach, part of a broader campaign targeting multiple U.S. agencies since December, remains under investigation, with Acuity implementing additional security measures and cooperating with law enforcement.

[Bleeping Computer](#)

## GitHub Status info: 8 issues

MARCH '24

Incidents that had a major impact – 2 incidents

**Total time of partial disruption – 2 hours 45 minutes**

Incidents that had a minor impact – 6 incidents

**Total time of degraded performance – 10 hours 39 minutes**

In its Status for March, GitHub experienced 8 issues, including 2 major incidents with over 2 hours of partial disruptions, primarily caused by authentication service degradation and permissions regression.

Also, there were 6 minor incidents totaling 10+ of degraded performance, affecting services like Pull Requests, Actions, and Copilot, with delays and elevated error rates peaking at 45% for some features.

---

# The software documentation platform is breached

Mintlify, a software documentation platform that assists developers in creating their documentation via accessing their customers' GitHub repositories, including AI, database, and fintech companies, experienced a security breach that exposed the GitHub tokens of 91 customers, potentially compromising private repositories.

The vulnerability, attributed to a flaw in its system, allowed attackers to gain unauthorized access using admin tokens, though there is no evidence of widespread access to customer repositories. Mintlify has revoked all GitHub token access, patched the vulnerability, implemented enhanced security measures, and advised users to change passwords, enable two-factor authentication, and review API key permissions.

The company continues to work with GitHub, customers, and third-party cybersecurity vendors to investigate and prevent future incidents.

*"The users have been notified, and we're working with GitHub to identify whether the tokens were used to access private repositories,"* stated in the company's advisory.

[HACK READ](#)

## GitHub Status info: 9 issues

FEBRUARY '24

Incidents that had a major impact – 2 incidents

**Total time of partial disruption – 2 hours 2 minutes**

Incidents that had a minor impact – 7 incidents

**Total time of degraded performance – 12 hours 18 minutes**

In February, GitHub faced 9 issues, including 2 major incidents with about 2 hours of partial disruptions, primarily impacting GitHub Packages, Webhooks, and Actions.

What's more, there were 7 minor incidents totaling around 12 hours of degraded performance, affecting services like Copilot, Git Operations, and dynamic content loading, with delays and intermittent errors reported across various functionalities.

## GitHub Status info: 9 issues

JANUARY '24

Incidents that had a major impact – 4 incidents

**Total time of partial disruption – 7 hours 7 minutes**

Incidents that had a minor impact – 5 incidents

**Total time of degraded performance – 99 hours 36 minutes**

In its Status for January 2024, GitHub reported 9 incidents, including 4 major ones causing around 7 hours of partial disruption, impacting services like Codespaces, live updates, and backend latency.

Another 5 issues were minor incidents that resulted in over 99 hours of degraded performance, with issues like delayed audit logs, avatar image failures, and port forwarding problems in Codespaces. Despite these challenges, the disruptions were mitigated through routing adjustments and system updates.

---

# A leaked GitHub token triggers Mercedes Source Code exposure

Due to a mishandled GitHub token, Mercedes-Benz source code was exposed to the public. A Mercedes-Benz employee leaked a GitHub token in their personal repository, granting unrestricted access to all source code on the company's GitHub Enterprise server.

The breach, which occurred on September 29, 2023, was discovered only on January 11, 2024, with the token being revoked two weeks later after RedHunt Labs alerted Mercedes. During the exposure, attackers could have accessed critical information such as API keys, design documents, database credentials, and other sensitive data, potentially causing financial, legal, and reputational damage.



*“We can confirm that source code containing an internal access token was published on a public GitHub repository by human error. This token gave access to a certain number of repositories, but not to the entire source code hosted at the Internal GitHub Enterprise Server. We have revoked the respective token and removed the public repository immediately. Customer data was not affected as our current analysis shows. We will continue to analyse this case according to our normal processes.” – Mercedes-Benz*

This incident underscores the severe risks associated with mishandled access tokens and highlights the need for stringent security protocols.

[Bleeping Computer](#) / [Security Week](#)

---

# High-severity vulnerability cases GitHub to rotate its keys

After patching a critical vulnerability tracked as CVE-2024-0200 with a CVSS score of 7.2 out of 10, GitHub rotates its keys.

A security flaw that GitHub in December 2023 allowed attackers with organization owner roles to access credentials within production containers via environment variables. The issue was reported through GitHub's Bug Bounty Program and promptly fixed, with credentials being rotated as a precaution.

GitHub confirmed no evidence of prior exploitation, attributing this confidence to the issue's uniqueness and internal telemetry.

However, the service provider reported that the customers using specific keys, such as GitHub Actions or Dependabot encryption keys, must import new public keys. Moreover, regular updates to public keys from the API are recommended for seamless future key changes.

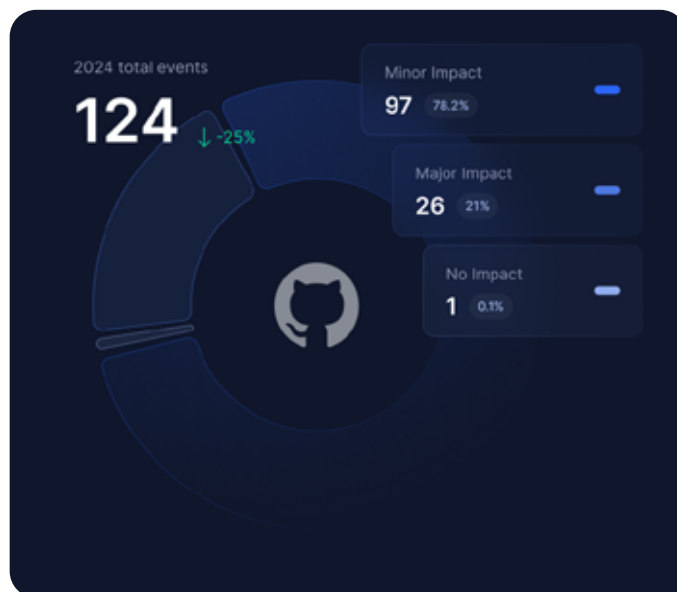
GitHub also addressed another vulnerability tracked as CVE-2024-0507, allowing privilege escalation through a Management Console editor role.

These incidents highlight GitHub's proactive approach to securing its platform, following previous key rotation and credential revocation cases earlier in the year.

# Takeaway

There were numerous incidents reported of how attackers abuse GitHub in their malicious schemes. Among the top malware that they were trying to spread using GitHub, were BeaverTail, Golssue, NjRat, Lumma Stealer, etc. However, GitHub was successful in its tries to stop malware distribution by deleting malicious accounts, repositories, and comments.

In 2024 GitHub experienced 124 incidents of different severity, which included partial disruptions, degraded service, and even an outage during which GitHub users couldn't access the service for over 30 minutes. Of other incidents, there were 26 incidents of partial disruption, totaling over 129 hours of it, and 97 cases of degraded performance with summed up time of issues counted for over 669 hours (which is more than 80 working days!).



Though GitHub takes all the necessary measures to provide safe and secure service, it is also the user's responsibility to take care of their data. Among the security best practices are periodic rotation of keys, regular repository scans, updates of the instance should GitHub release new patches, backup, and DR procedures.



Atlassian 2024 report (BitBucket & Jira)

# The State of Atlassian Threat Landscape - 2024 In Review

14

major impact issues

21<sub>hrs</sub>

of disruption



GitProtect  
by Xopero ONE

# The State of Atlassian Threat Landscape – 2024 In Review

The year 2024 was full of incidents, hacker attacks, and security flaws that were used in the wild. Were there any incidents related to the Atlassian ecosystem? Outages? Security flaws? We've decided to analyze it.

In comparison to the previous year, the number of incidents in Jira grew by 44%, and when we compare it to 2022 the percentage is even higher – 63% – in 2022 there were reported 59 incidents, in 2023 – 75, and in 2024 their number grew to 132 incidents.

## Jira recorded incidents during the year

# 132

incidents total

■ 2024  
■ 2023



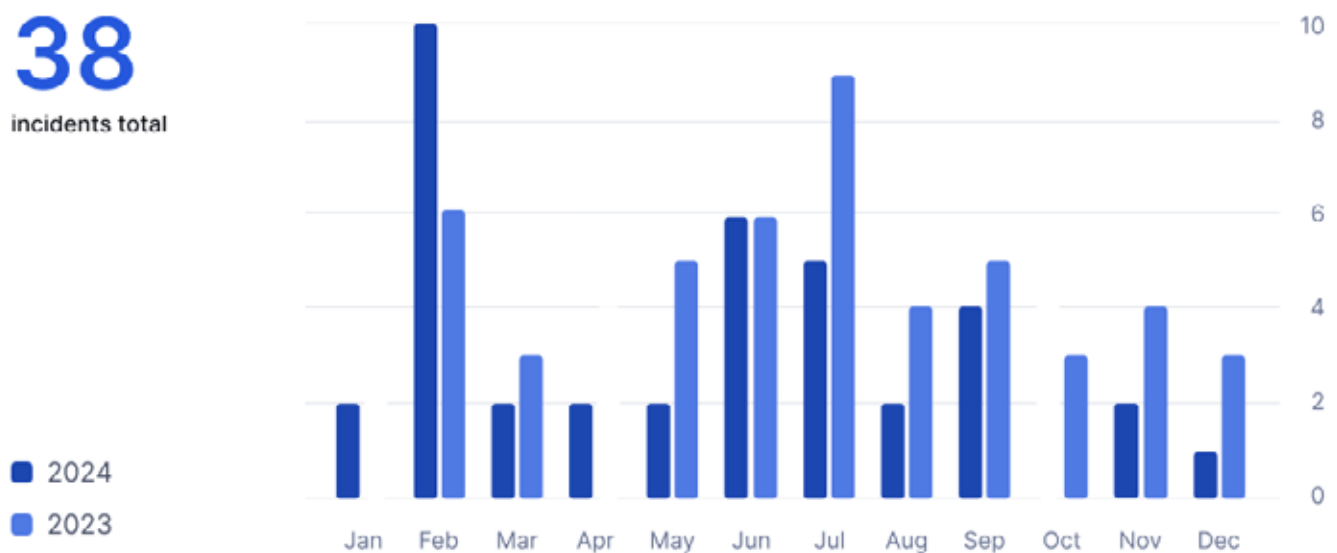
GitProtect.io

When it comes to Bitbucket, in 2024 there were 38 incidents reported, where we can see a 20% decrease compared to 2023 with 48 incidents reported.

### Bitbucket recorded incidents during the year

**38**

incidents total



GitProtect.io

Well, let's dive into the topic and see which cybersecurity incidents that happened in 2024 were mentioning Atlassian products...

Atlassian Status info for Bitbucket – 1 incident

DECEMBER '24

**Total time of incidents to last – 44 minutes**

Atlassian Status for Jira – 11 incidents

**Total time of incidents to last – 81 hours 22 minutes**

In December, Bitbucket users might face a 44-minute incident that involved issues with attachments in Git LFS. There was a degradation of performance in viewing previews, downloading, and uploading large files.

When it comes to Jira, Atlassian faced 11 incidents, totaling more than 81 hours of degraded performance. Among them, several service disruptions affected Jira, Confluence, and JSM. Also, there were failures in Automation rules, the “*Send Web Request*” action, degraded AI features, and problems with Jira loading, comments, and attachments.

---

# North Korean hackers target software developers in their malware campaign

A North Korean hacking campaign, “*Contagious Interview*,” aimed at software developers through fake job postings to deploy malware such as BeaverTail and OtterCookie.

The OtterCookie malware, introduced in September 2023, is delivered via loaders infecting Node.js projects, npm packages, or files built as Qt or Electron applications hosted on platforms like Bitbucket and GitHub. These loaders execute malicious JavaScript code, enabling data theft, including cryptocurrency keys, documents, and clipboard content.

OtterCookie establishes secure links to command-and-control servers and supports reconnaissance and infiltration commands, facilitating deeper attacks.

Atlassian’s Bitbucket, frequently used in these infection chains, highlights the need for vigilance in software supply chain security among developers.

[Techzine](#)

Atlassian Status info for Bitbucket – 2 incident

NOVEMBER '24

**Total time of incidents to last – 2 h 25 min**

Atlassian Status info for Jira, JWM, JSM – 14 incidents

**Total time of incidents to last – 193 hours 34 minutes**

Atlassian reported multiple incidents and maintenance activities affecting its products in November. Bitbucket experienced over 2 hours of degraded performance due to a missing reCAPTCHA token preventing user invitations.

Jira, Jira Work Management, and Jira Service Management were impacted by 14 incidents, resulting in a total disruption time of more than 193 hours, including a 68-hour automation issue.

---

# Schneider Electric's 400K rows of data are stolen in a dev platform breach

Schneider Electric confirmed a breach involving its internal project tracking platform hosted in an isolated environment. The threat actor, known as “Grep,” claims to have accessed the company’s Jira server using exposed credentials and stole 40GB of data, including 400K rows of user information, 75K unique email addresses, and other critical project data. The stolen information reportedly includes details about projects, issues, and plugins, and the attackers have demanded \$125,000 to prevent data leakage.

Schneider Electric stated that its products and services remain unaffected and that its Global Incident Response team is actively investigating the incident. The attackers, part of a newly formed hacking group named Hellcat, initially sought to extort the company but have also hinted at developing ransomware for future attacks. This follows a previous breach earlier in 2023 when Schneider Electric’s “*Sustainability Business*” division was targeted in a Cactus ransomware attack.

[Bleeping Computer](#)

---

# Atlassian patches security flaws in Bitbucket, Jira, Confluence and other tools

In November Atlassian addressed and patched 20 security flaws in its products, including a high-severity vulnerability to prevent potential DoS states and code execution attacks. Such flaws can often be exploited without authentication using crafted signatures or HTTP/2 requests, although detailed exploitation methods have not been disclosed.

Admins are urged to promptly install secure versions, as there are no current reports of active exploitation but also no Indicators of Compromise (IoCs) provided to detect attacks.

Protected versions include updates for Bitbucket, Confluence, Jira, Jira Service Management, Bamboo, Crowd, and Sourcetree for macOS and Windows.

[Heise Online](#)

Atlassian Status info for Bitbucket

OCTOBER '24

**– No incidents reported for this time**

Atlassian Status info for Jira, JWM, JSM – 19 incidents

**Total time of incidents to last – 428 h 44 min**

In October, there were no reported incidents for Bitbucket. However there were some for Jira, JWM, and JSM. The tools were affected by 19 incidents, resulting in a total disruption time of over 425 hours.

Major issues included migration failures, automation rule disruptions, and degraded performance in user management and project access. Other problems involved delays in development information syncing and intermittent errors impacting user functionality.

---

# Atlassian addresses 6 high-severity vulnerabilities in Bitbucket, Jira and Confluence

In its [October security bulletin](#), Atlassian reported patching 6 high-severity vulnerabilities across Bitbucket, Confluence, and Jira Service Management Data Center and Server products. These include CVE-2024-21147 with a CVSS score of 7.4, a Java Runtime Environment flaw leading to unauthorized access in Bitbucket, and multiple issues in Confluence, such as Moment.js library vulnerabilities, an XSS bug, and an Apache Commons Configuration flaw.

Jira Service Management updates addressed CVE-2024-7254 with a CVSS score of 7.5, a Protobuf buffer overflow that could disrupt service availability.

While none of these vulnerabilities were known to have been exploited in the wild, Atlassian urged users to apply the updates promptly.

[Security Week](#)

---

## Cisco's data breach – confirmed

On October 14th, 2024, a hacker, known as IntelBroker, announced that he managed to breach Cisco and steal some company's data, including Jira tickets, GitHub and SonarQube projects, source code, confidential documents, etc.

Cisco's investigation determined its systems were not breached; instead, the data was accessed from a public-facing DevHub environment containing customer resources like scripts and source code. Cisco has found no evidence of sensitive personal or financial data being compromised but is continuing its investigation.

In response, the company has disabled public access to the affected website.

[Security Week](#)

Atlassian Status info for Bitbucket – 4 incidents

SEPTEMBER '24

**Total time of incidents to last – 3 hours 5 minutes**

Atlassian Status info for Jira, JWM, JSM – 19 incidents

**Total time of incidents to last – 72 hours 55 minutes**

Atlassian experienced 23 incidents in total for Bitbucket, Jira, JSM, and JWM. Four incidents happened in Bitbucket's infrastructure, totaling around 3 hours of their occurrence, including issues involving reCAPTCHA errors during signup and performance degradation.

The number of incidents impacting Jira, JWM, and JSM was 19, which lasted in total almost 73 hours. There were some issues with workflow provisioning, asset loading, automation rules, and email replies creating new tickets instead of comments.

---

# Six high-severity Atlassian security flaws are patched

September brought [6 patches for high-severity vulnerabilities](#) in the Atlassian ecosystem. The service provider addressed security flaws in Bitbucket, Jira, Jira Service Management, Confluence, BambooCrowd, and Sourcetree, which could allow attackers to create DoS states or execute malicious code. Exploits may involve prepared signatures or special HTTP/2 requests and, in some cases, may not require authentication.

Although there were no reports of active exploitation in the wild, Admins were urged to apply the updates promptly to secure their systems.

## [Security Week](#)

Atlassian Status info for Bitbucket – 3 incidents

AUGUST '24

**Total time of incidents to last – 3 hours 36 minutes**

Atlassian Status info for Jira, JWM, JSM – 15 incidents

**Total time of incidents to last – 313 hours 2 minutes**

In August, Atlassian reported 3 incidents in Bitbucket lasting in total of 3+ hours, and 15 incidents in Jira, JWM, and JSM that lasted for over 310 hours.

The issues ranged from login timeouts and functionality degradation in workflows, backlogs, and assets in Jira to degraded performance with the Bitbucket website.

Atlassian Status info for Bitbucket – 5 incidents

JULY '24

**Total time of incidents to last – 14 hours 09**

Atlassian Status info for Jira, JWM, JSM – 5 incidents

**Total time of incidents to last – 29 hours 33 minutes**

Atlassian reported 5 incidents for Bitbucket in July, totaling over 14 hours of effect, with two major incidents causing around 30 minutes of downtime and around 90 minutes of Bitbucket pipeline disruption.

Regarding Jira, JWM, and JSM, Atlassian had to deal with 5 incidents as well which resulted in over 29 hours of effect. The incidents included degraded performance in services and SLA data loss.

---

# Hackers exploit critical Atlassian flaw to connect servers in mining networks

According to cybersecurity researchers from Trend Micro, hackers could actively exploit a critical vulnerability, tracked as CVE-2023-22527 with a CVSS score of 10 (out of 10!), in Atlassian's Confluence Data Center and Server products.

The security flaw allows Remote Code Execution (RCE) and poses serious risks to affected systems. The vulnerability, present in older versions, has been actively used by threat actors for crypto-jacking, with a surge in attacks observed from June to July 2024.

Attackers deploy sophisticated scripts to disable security measures, spread via SSH, and establish unauthorized cryptocurrency mining, compromising system integrity and resources.

Cybersecurity experts recommend immediately updating Confluence to mitigate the vulnerability and prevent further exploitation.

[gbhackers](#)

Atlassian Status info for Bitbucket – 6 incidents

JUNE '24

**Total time of incidents to last– 52 hours 27 minutes**

Atlassian Status info for Jira, JWM, JSM – 6 incidents

**Total time of incidents to last – 226 hours 26 minutes**

In June, Atlassian reported 6 incidents for Bitbucket, which lasted in total for over 50 hours. The incidents included degradation in the work of pipelines and error responses across the product.

As for Jira, JWM, and JSM, there were also 6 incidents recorded, totaling over 226 hours of disruptions. Among the cases that Atlassian was dealing with were slow performance, issues with the option to copy production data to sandbox, etc.



# Club Penguin fans stole 2.5 GB of internal corporate data in Disney Confluence server breach

Club Penguin fans exploited Disney's Confluence server to access old internal game data but inadvertently stole 2.5 GB of sensitive corporate information, including developer tools, internal infrastructure, advertising plans, and business documentation.

The image shows a character sheet for SENSEI, a penguin character. The sheet is titled "Character Sheets" and "Modified June 11, 2014 10:02 AM". It includes a character illustration, a speech bubble, and various sections of text.

**CHARACTER FACTS**

**CHARACTER**  
Character Title: Master of the Elements, Master of Card-Jitsu, Teacher of Ninjas  
Character Role: The Nurturing Teacher  
Character Goal: Maintaining balance over the Card-Jitsu elements Fire, Water, Snow.  
Character Description: The wise and powerful ninja SENSEI mentors his students in the ancient art of Card-Jitsu to master the elements of Fire, Water and Snow. Curious and quick-thinking, SENSEI is adaptable and accepting, unless one of his core values is threatened.

**UNIQUE ITEMS/APPEARANCE**

- SENSEI hat
- Eye beard
- Gray bushy eyebrows
- Eye wrinkle

**FAVORITE THINGS**

Drinks: Tea  
Foods: Rice with hot sauce  
Game: Card-Jitsu  
Locale: Dojo  
Possessions: His walking staff (Tusk's broken tusk)

**DEFINING MOMENT OF AWESOME**  
Leading ninjas to face Tusk, the ultimate Master of Snow in battle at Tusk's cave.

**FAMOUS DEEDS**

- Built the Dojo long ago, before vanishing into the Wilderness to learn the ancient art of Card-Jitsu.
- Can summon and control the elements. Is able to manifest them at will.
- Appeared out of nowhere when lightning struck the Dojo in 2008. This was a sign from the elements that it was time to return.

**TOP 5 QUIRKS**

1. Occasionally uses "deep" Confucius-like statements.
2. He trained in the ancient art of Card-Jitsu as a young boy.
3. He has always carried the guilt of not saving his childhood friend, Tusk in the avalanche.
4. Carries Tusk's broken tusk as his walking stick.
5. Has a secret gray puffle that no one knows about.

**RANDOM FACTS**  
Elder of all the penguins on the island, and the only gray penguin.

**FRIENDS AND COMPANIONS**  
All ninja students of Card-Jitsu

**CHARACTER REACTIONS**

**GETS SCARED**  
For a moment, he smiles placidly. The lake doesn't even ripple. Then a low chuckle, followed by, "Well done. I have never been more scared in my life."

**VERY HUNGRY**  
He rebukes in food. He will consume tea and rice without hesitation. He is excellent dinner company.

**GIVEN A GIFT**  
SENSEI would bow deeply and accept them gladly. He would remark, "Your treasures are those that come freely from the heart. Giving is the gift."

**COMPLIMENTED**  
His expression would not change, but he might bow slightly. He would seek to reflect the compliment back, such as, "A great teacher is only realized in the greatness of his students."

**GETS BORED**  
Never gets bored. With age has come infinite patience and a curiosity that causes him to find interest in everything.

**GETS ANGRY**  
A fierce storm comes over his features. He stands to full height and the elements reside before him.

**GETS SAD**  
Head lowers slightly, and he strokes his beard. He breaths a deep troubled breath and moans. He accepts and moves on, never revealing the turmoil within. "So it is, my students. So it is."

**STUBS "TOE"**  
He would acknowledge the pain and let it pass. "Thank you are my old friend, thank you are."

**CHARACTER TRAITS**

**EMBODIES**  
Mystery, Discovery, Adventure, Power

**TRUE COLORS**  
Gold, Orange, Blue, Green

**Supporting**  
Guilt  
Kind  
Playful

©Disney Do not duplicate or distribute

Page 117

Source: Bleeping Computer – SENSEI character sheet from leaked documents



*“Lot more files here including internal api endpoints and credentials for things like S3 buckets,” is reported in the [Bleeping Computer](#) article.*

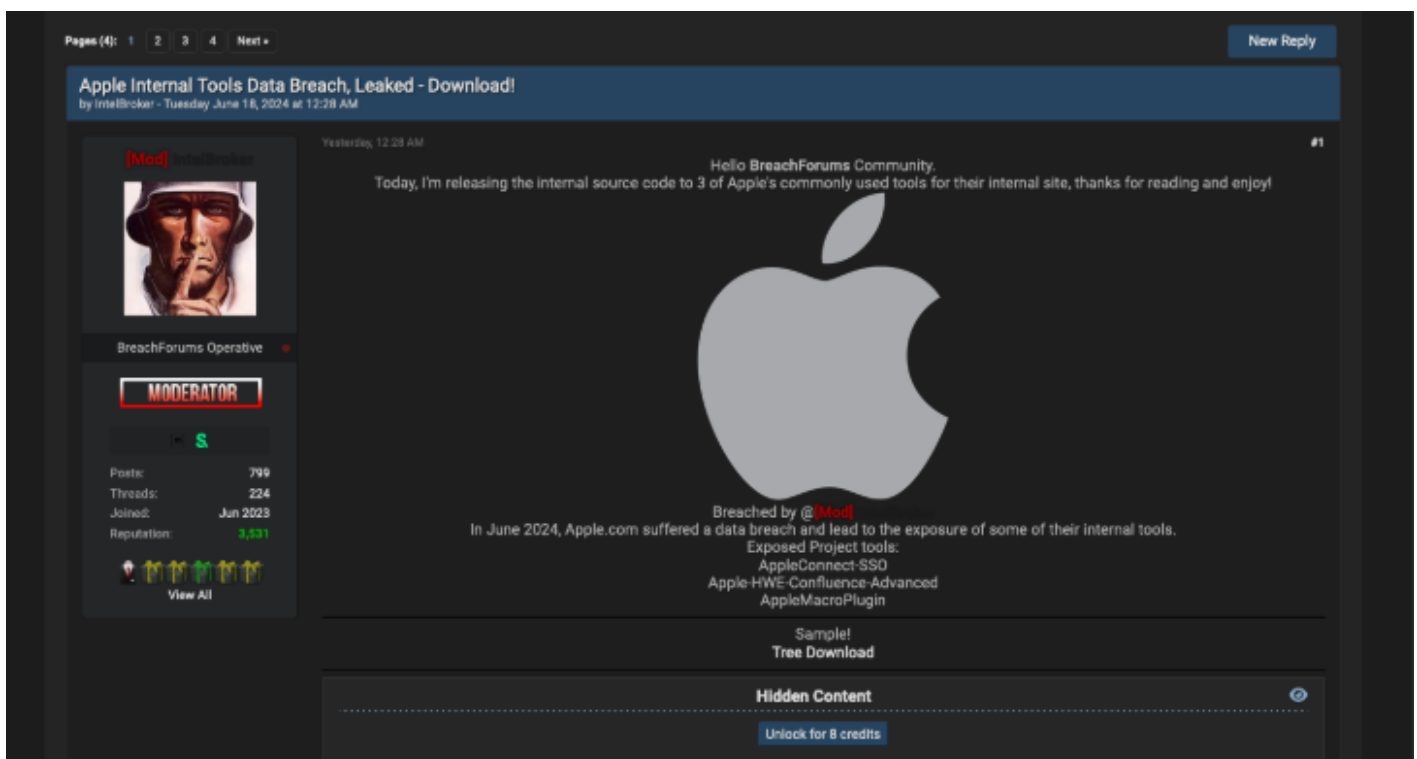
The breach occurred using previously exposed credentials and included internal API endpoints, S3 bucket credentials, and links to developer resources, potentially increasing Disney’s exposure to further attacks.

Atlassian’s Confluence platform, central to Disney’s documentation, was the target of the breach, highlighting the risks of improperly secured or outdated credentials in enterprise environments. While the initial focus of the attackers was nostalgic Club Penguin data, the newer data with information from 2024 was also circulating on Discord.

[Bleeping Computer](#)

# A hacker claims to have breached Apple

A threat actor known as IntelBroker claims to have leaked internal Apple code in a data breach in June 2024, targeting tools used within Apple's corporate network, including AppleConnect-SSO.



Source: Heise Online

The leaked data, analyzed by cybersecurity firm AHCTS, contains proprietary plugins and configurations for integrating Apple's authentication systems with Atlassian Jira and Confluence, posing significant cybersecurity risks but not affecting end-user products or services.

Atlassian Status info for Bitbucket – 2 incidents

MAY '24

**Total time of incidents to last – 3 hours 13 minutes**

Atlassian Status info for Jira, JWM, JSM – 10 incidents

**Total time of incidents to last – 485 hours 56 minutes**

In May, Atlassian reported 2 incidents for Bitbucket, resulting in 3 hours and 13 minutes of degraded performance in Git LFS operations and Bitbucket pipelines, and 10 incidents occurred in Jira services, totaling over 485 hours of partial disruptions, primarily driven by scheduled infrastructure maintenance in Brazil that caused extended outages for Free edition users.

[Forbes](#) / [Heise Online](#)

# Hackers use authentication secrets leaked as plaintext in Atlassian Bitbucket artifact objects in their malicious activity

Threat actors exploited plaintext AWS authentication secrets exposed in Atlassian Bitbucket artifact objects to breach accounts. A cyber security firm, Mandiant, discovered this vulnerability while investigating AWS access abuse, revealing that sensitive data, such as secured variables, could unintentionally appear in artifact files during CI/CD processes.

Despite Bitbucket's design to store secured variables in encrypted form, developers may unknowingly export these secrets as plaintext to artifact files via commands like `printenv`. If these artifacts are stored in public repositories, attackers can access and misuse the plaintext secrets.

## Repository variables

Environment variables added on the repository level can be accessed by any users with push permissions in the repository. To access a variable, put the `$` symbol in front of its name. For example, access `AWS_SECRET` by using `$AWS_SECRET`.

[Learn more about repository variables.](#)

Repository variables override variables added on the workspace level. [View workspace variables](#)

If you want the variable to be stored unencrypted and shown in plain text in the logs, unsecure it by unchecking the checkbox.

Name	Value	<input checked="" type="checkbox"/> Secured	Add
AWS_DEFAULT_REGION	us-east-1	<input checked="" type="checkbox"/>	 
AWS_SECRET_ACCESS_KEY	*****	<input checked="" type="checkbox"/>	 
AWS_ACCESS_KEY_ID	*****	<input checked="" type="checkbox"/>	 

*Source: Bleeping Computer – How to store variables secured in Bitbucket*

Mandiant noted that this often occurs when developers misconfigure pipeline scripts or use artifacts for debugging without considering security implications.

To mitigate these risks, developers should use dedicated secret management tools, carefully review artifact content, and implement code scanning to detect and remove sensitive data before deployment.

[Bleeping Computer](#) / [The Cyber Express](#)

Atlassian Status info for Bitbucket – 2 incidents

APRIL '24

**Total time of incidents to last – 8 hours 19 minutes**

Atlassian Status info for Jira, JWM, JSM – 5 incidents

**Total time of incidents to last – 13 hours 48 min**

Two incidents were reported in Bitbucket in April. There were issues with Admin portal feature access and pipelines, totaling over 8 hours of lasting incidents.

Jira services experienced issues for over 13 hours in total – there were 5 incidents, including intermittent errors with linking issues in the issue view, scheduled maintenance, and interruptions with installing apps on user's instances.

---

# Critical Atlassian flaw allows Linux variant of cerber ransomware deployment

Threat actors exploit CVE-2023-22518, with the CVSS score of 9.1 out of 10 maximum, in the wild. This critical vulnerability in Atlassian Confluence servers, to deploy the Cerber ransomware variant. The vulnerability allows attackers to reset Confluence, create an admin account, and take control of the system, leading to potential full compromise.

Using this access, attackers can install the Effluence web shell plugin to execute commands and deploy the ransomware payload, which encrypts files with a .LOCK3D extension and drops ransom notes without exfiltrating data. Cerber's primary payload, written in C++, acts as a loader for other malware retrieved from a command-and-control server, highlighting a rare but sophisticated use of pure C++ in modern ransomware.

The attack's efficacy is often limited, as the ransomware typically only encrypts Confluence-specific data, which may be backed up in well-configured systems. This limitation reduces the likelihood of victims paying ransom. Which highlights the importance of having a reliable backup provider, like GitProtect.io.

The rise of Cerber coincides with the emergence of other ransomware families and variants exploiting leaked source codes, such as LockBit, to create customized threats. Security researchers emphasize the importance of strong security measures, regular patching, and fostering a cybersecurity-aware culture to mitigate such evolving threats.

[The Hacker News](#)

Atlassian Status info for Bitbucket – 2 incidents

MARCH '24

**Total time of incidents to last – 1 hours 28 minutes**

Atlassian Status info for Jira, JWM, JSM – 7 incidents

**Total time of incidents to last – 24 hours 35 minutes**

In March, Atlassian reported 2 incidents for Bitbucket, including issues with Admin portal feature access and the Cloud service degradation that lasted for over an hour.

In Jira services, there were reported 7 incidents that resulted in over 24 hours of issues. The incidents included the impossibility of scheduling data residency migrations, issues with creating issues and transitions, etc.

Atlassian Status info for Bitbucket – 10 incidents

FEBRUARY '24

**Total time of incidents to last – 22 hours 10 minutes**

Atlassian Status info for Jira, JWM, JSM – 15 incidents

**Total time of incidents to last – 128 hours 58 minutes**

---

Atlassian reported 10 incidents for Bitbucket and 15 incidents for Jira services in February 2024.

Bitbucket experienced in total of 22+ hours of service issues, including service disruptions, fails in user searches, pipelines stuck in a pending state, and some authentication errors.

Jira, JSM, and JWM saw around 129 hours of issues, which included issues with automation and connecting apps, SLA issues, 503 errors, and service disruptions.

# Cloudflare data breach due to auth token stolen in Okta attack

The bad actor managed to breach Cloudflare's internal Atlassian server and access the company's Jira bug database, Bitbucket source code management system, and Confluence wiki.

later established persistence using the ScriptRunner plugin.



The attacker initially gained access to the company's systems on November 14, 2023, using credentials stolen during an earlier Okta breach and



*"They then returned on November 22 and established persistent access to our Atlassian server using ScriptRunner for Jira, gained access to our source code management system (which uses Atlassian Bitbucket), and tried, unsuccessfully, to access a console server that had access to the data center that Cloudflare had not yet put into production in São Paulo, Brazil," [explained Cloudflare](#).*

Once the activity on November 23 was detected, Cloudflare cut off the hacker's access by November 24 and began a detailed forensic investigation on November 26. Remediation included rotating over 5K production credentials, reimaging nearly 4,900 systems, segmenting test environments, and ensuring

the security of a São Paulo data center targeted during the breach, which was unsuccessful.

Despite the extensive measures, Cloudflare confirmed no customer data, systems, or global network configurations were affected. The attackers appeared focused on gathering information about Cloudflare's network architecture and security to establish a deeper foothold.

[Bleeping Computer](#) / [The Hacker News](#) / [Dark Reading](#)

Atlassian Status info for Bitbucket – 2 incidents

JANUARY '24

**Total time of incidents to last – 5 hours**

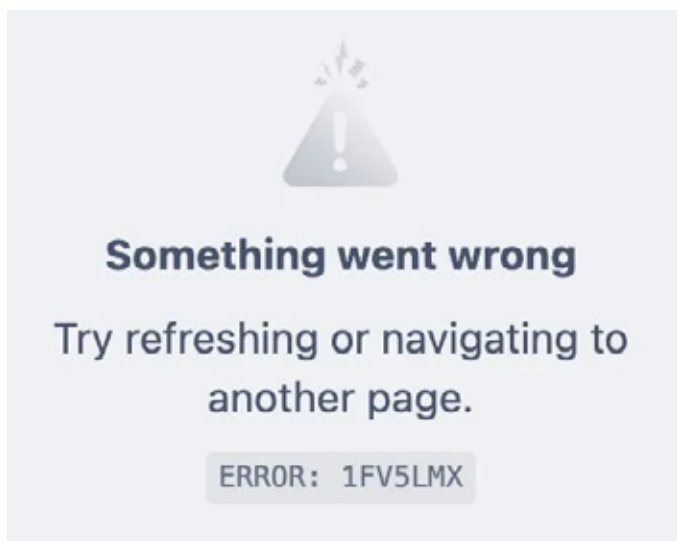
Atlassian Status info for Jira, JWM, JSM – 7 incidents

**Total time of incidents to last – 135 hours 26 minutes**

In January, Atlassian reported 2 incidents for Bitbucket, totaling 5 hours of impact, including performance degradation in Forge app invocations and an outage in Atlassian Intelligence functionality.

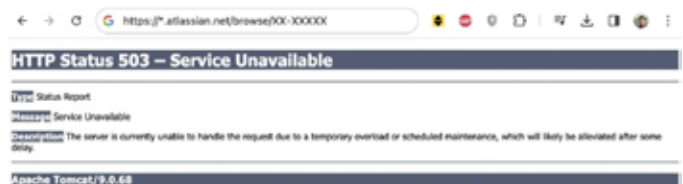
For Jira, JWM, and JSM, there were 7 incidents leading to over 135 hours of lasting incidents, including issues in sandbox sites, SCIM provisioning syncs, and webhook functionality, among others.

The most noticeable incident that affected Jira users was an outage that occurred on January 15th and lasted for over 3 hours.



The error message that users may see on Jira panels and widgets.

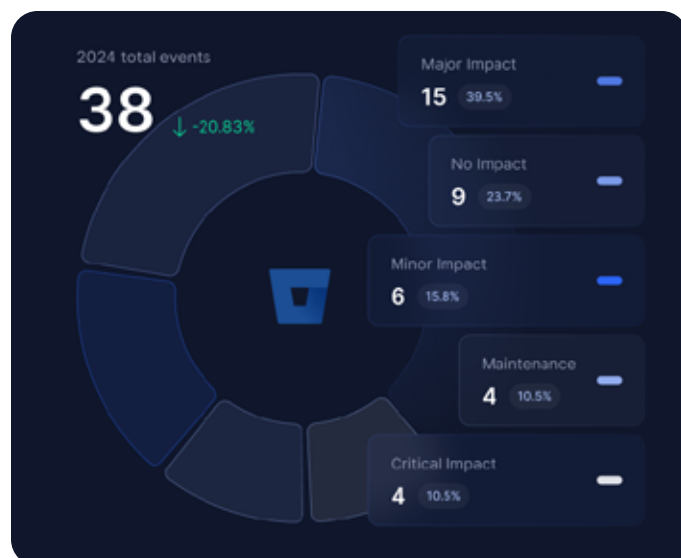
On the Jira Status page, Atlassian explained the incident was triggered by *“a scheduled database upgrade within the central service that supports the Atlassian Marketplace.”*



Source: [Bleeping Computer](#) – HTTP 503 error message that customers might see when accessing Jira Cloud services

# Takeaway

In 2024 Bitbucket reported 38 incidents which lasted in total for over 110 hours while Jira services, including Jira Software, Jira Service Management and Jira Work Management experienced in total of 132 incidents that lasted for about 2131 hours... which makes around 266 working days!



Cybercriminals are becoming more aggressive in their attacks, and we can see it not only in the example of Atlassian but also in GitHub, GitLab, and Azure DevOps.

It means that organizations should take security more seriously and implement security mechanisms to make sure that their infrastructure as well as data is safe.



Such measures may include in-time updates, implementation of the least privilege principle, and backup of critical DevOps and project management data.

 GitLab 2024 report

# The State of GitLab Threat Landscape - 2024 In Review

7

service disruption incidents

798<sub>hrs</sub>

of disruption



GitProtect  
by Xopero ONE

# The State of GitLab Threat Landscape – 2024 In Review

Let's move on with our research on the DevOps threat landscape in 2024. Let's see which security incidents and vulnerabilities GitLab users faced in the previous year.

In 2023, GitLab reported 76 incidents on their Status page, this year the number of incidents grew by almost 21% and compiled 97 incidents in total.

## GitLab recorded incidents during the year

**97**

incidents total

■ 2024  
■ 2023



GitProtect.io

## GitLab Status info: 10 issues

DECEMBER '24

Incidents that had a major impact – 7 incidents

**Total time of partial service disruption – 130 hours 13 minutes**

Incidents that had a minor impact – 2 incidents

**Total time of degraded performance – 17 hours 44 minutes**

Security issue – 1 incident

**Total time of the security issue – 3848 hours 5 minutes**

In December, GitLab faced 10 incidents, including 7 with a major impact, resulting in over 130 hours of partial service disruption.

Users might encounter intermittent errors, timeouts, and disruptions in CI/CD services, Git operations, and website functionality across multiple locations. The most significant incident, lasting 75 hours and 2 minutes, involved intermittent timeouts for requests from some Utah IP addresses.

Another major incident, spanning 26 hours and 16 minutes, caused errors across GitLab.com, affecting websites, pages, and background processing. Other disruptions included CI/CD runner failures and GitLab Duo chat issues.

## GitLab Status info: total 8 incidents

NOVEMBER '24

Incidents that had a critical impact – 1 incident

**Total time of service disruption – 1 hour**

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption – 33 hours 17 min**

Incidents that had a minor impact – 3 incidents

**Total time of degraded performance – 18 hours 46 minutes**

Planned Maintenance – 2 incidents

**Total time of planned maintenance – 66 hours**

GitLab users experienced 8 incidents, including a critical one that caused customer portal disruption for an hour.

Also, there were 2 major incidents that resulted in partial service disruptions, totaling over 33 hours and impacting the Jira Cloud app installation and the group API.

Other three minor incidents led to degraded performance for nearly 19 hours, affecting features like merge requests, user preferences, and Duo Chat.

---

# GitLab vulnerability exploited in major Byte Federal cyberattack

The personal information of around 58K customers of Byte Federal, a leading Bitcoin ATM operator in the USA, was exposed by cybercriminals.

On November 18, 2024, Byte Federal's security team discovered that hackers used a GitLab security flaw to breach the company. The hack exposed personal information, including names, dates of birth, residential addresses, phone numbers and email addresses, Social Security Numbers, transaction histories, photos, etc.

To address the issue the company promptly secured its systems and isolated the threat actor, who managed to access one of Byte Federal's servers. Though the exact nature of the GitLab vulnerability the attackers used wasn't identified, the organization conducted a complete reset of all its customers' accounts, revoked access tokens, and updated internal passwords.

Also, Byte Federal urged all the affected customers to be attentive to any suspicious activity and report to the authorities in any case of fraudulent incidents.

[iZoologic](#)

---

# Patching security gaps against unauthorized access

The total of six security flaws, GitLab developers patched to protect GitLab CE/EE users in November. Among them, there were vulnerabilities of medium severity and one of a high severity level – 8.5 of 10. Using the vulnerability, tracked as [CVE-2024-9693](#), an attacker could get unauthorized access to the Kubernetes agent in a cluster in some specific configurations.

## Among less severe security flaws that were patched, were:

- CVE-2024-7404 with a CVSS score of 6.8, which allowed the hacker to gain full API access via the device OAuth flow.
- DoS issue with the CVSS score of 6.5 which could take place upon importing maliciously crafted content via Fogbugz importer.
- CVE-2024-8648, with a severity of 6.1, that could permit the threat actor to inject malicious JavaScript code into the Analytics Dashboards via a specifically created URL.
- CVE-2024-8180 with a CVSS score of 5.4 – the HTML injection in this vulnerability might lead to XSS on the self-hosted instances.
- CVE-2024-10240 with a CVSS score of 5.3 which allowed an unauthorized user under some specific circumstances to read some info about an MR in private projects.

GitLab advised its users to update their GitLab Community Edition and Enterprise Edition to mitigate the risks that the mentioned security flaws could expose.

[Heise online](#)

**GitLab Status info: total 9 incidents**

OCTOBER '24

Incidents that had a critical impact – 1 incident

**Total time of service disruption – 11 hours 23 minutes**

Incidents that had a major impact – 6 incidents

**Total time of partial service disruption – 83 hours 49 minutes**

Incidents that had a minor impact – 1 incident

**Total time of degraded performance – 1 hour 15 minutes**

Planned Maintenance – 2 incidents

**Total time of planned maintenance – 77 hours**

There were 9 incidents of different impact that the GitLab had to deal with in October. One of them, the critical one, lasted for roughly 11 hours and affected pipelines using SaaS Runners that were stuck on pending status.

Other 6 less major incidents caused partial disruptions totaling 80+ hours and impacting areas such as pipeline processing, self-managed runners, and Sidekiq job delays.

The rest of the minor incidents didn't have much impact on GitLab service operation for GitLab users.

---

# Critical arbitrary branch execution flaw in GitLab

[Multiple security flaws](#) were threatening GitLab Community Edition and Enterprise Edition users that the service provider successfully addressed in November 2024. There were 8 vulnerabilities, 5 of which were of high or critical severity.

The security flaw tracked as CVE-2024-9164 with a CVSS score of 9.6 might permit unauthorized users to run [Continuous Integration and Continuous Delivery pipelines](#) on any branch of a repo.

Another high severity vuln, tracked as CVE-2024-8970 with the CVSS score of 8.2, could permit the threat actor to trigger a CI/CD pipeline as another user.

## Among other security flaws that GitLab addressed were:

- CVE-2024-8977 the CVSS score of which is 8.2 – the SSRF vulnerability in the Analytics Dashboards which could leave users unsafe to SSRF attacks.
- CVE-2024-9631 with a CVSS score of 7.5, which could lead to slowing down the viewing diffs of MR with conflicts.
- CVE-2024-6530 with a severity score of 7.3, which is an HTML injection vulnerability in the OAuth page permitting cross-site scripting when a user tries to authorize through OAuth.

- CVE-2024-9623 with a CVSS score of 4.9 that permits deploying keys for pushing to the archived repository.
- CVE-2024-5005, the severity score of which is 4.3, using which unauthorized users are able to disclose project templates via the API.
- CVE-2024-9596, with a low severity of 3.7, which could permit the bad actor to identify the GitLab version number of the GitLab instance.

GitLab “*strongly*” recommended its users to update the affected installations to make sure that their data is safe.

[Bleeping Computer](#) / [The Hacker News](#)

## GitLab Status info: total 17 incidents

SEPTEMBER '24

Incidents that had a critical impact – 2 incidents

**Total time of service disruption – 1 hour 28 minutes**

Incidents that had a major impact – 3 incidents

**Total time of partial service disruption – 25 hours 13 minutes**

Incidents that had a minor impact – 4 incidents

**Total time of degraded performance – 270 hours 1 minute**

Operational issues – 5 incidents

**Total time of operational issues – 122 hours 44 min**

Planned Maintenance – 3 incidents

**Total time of planned maintenance – 74 hours**

In September, GitLab experienced 17 incidents with different impact levels. Two incidents recognized by GitLab as critical, lasting for more than an hour in total, resulted in connectivity issues across multiple GitLab services and downtime of the Customer Portal.

Other 3 less severe incidents caused partial service disruptions, totaling 25+ hours, affected APIs, Git operations, and project accessibility. Four minor incidents led to degraded performance for a combined 270 hours, with the longest issue involving errors with the “*View job currently using resource*” button.

# Authentication Bypass Bug is threatening GitLab CE and EE users

A critical authentication bypass vulnerability with a CVSS score of 10 (out of 10!) was patched in GitLab Community Edition (CE) and Enterprise Edition (EE). The security flaw tracked as [CVE-2024-45409](#) was found in the ruby-saml library. Caused by improper signature verification of SAML responses, the vulnerability could allow attackers to log in as arbitrary users. The issue also affects omniauth-saml, which has been updated to version 2.2.1, while GitLab has patched multiple versions, including 17.3.3 and 16.11.10.



*“An unauthenticated attacker with access to any signed SAML document (by the IdP) can thus forge a SAML Response/Assertion with arbitrary content... This would allow the attacker to log in as arbitrary user within the vulnerable systems,” states the [security advisory](#).*

To mitigate risks, GitLab recommends enabling two-factor authentication (2FA) and disallowing SAML two-factor bypass. While there is no confirmation of active exploitation, GitLab has provided indicators of potential abuse, such as specific SAML-related log events.

This update comes as the U.S. CISA adds new vulnerabilities, including a critical flaw in Apache HugeGraph-Server (CVE-2024-27348), to its Known Exploited Vulnerabilities catalog, urging agencies to have remediated them by October 9, 2024.

[The Hacker News](#)

[Help Net Security](#)

[Dark Reading](#)

## GitLab Status info: total 6 incidents

AUGUST '24

Incidents that had a major impact – 4 incidents

**Total time of partial service disruption – 50 hours 27 minutes**

Incidents that had a minor impact – 1 incident

**Total time of degraded performance – 1 hour 22 minutes**

Operational issues – 1 incident

**Total time of operational issues – 3 hours 32 minutes**

In August, GitLab experienced six incidents, including four major ones with a total partial service disruption time of about 50 hours. The most significant incident lasted around 27 hours – during that time GitLab users might have had some issues in the documentation search. Other issues included CI/CD token allowlist failures and Duo AI feature malfunctions.

---

# Critical pipeline execution vulnerability is patched – urgent update needed

Seventeen security vulnerabilities, including 4 high-severity, 11 medium-severity, and 2 of low-severity, were detected and patched by the GitLab DevOps team in August. The most critical issue, tracked as CVE-2024-6678 and a CVSS score of 9.9, might allow an attacker to run pipeline jobs as an arbitrary user.

Let's not forget that GitLab pipelines, which are part of [GitLab CI/CD](#), are automated workflows that DevOps use to build, test, and deploy their code. If a hacker gains access to them, it might lead to data loss.

In its [alert](#), GitLab warns that *“An issue has been discovered in GitLab EE affecting all versions starting from 11.1 before 17.1.7, 17.2 before 17.2.5, and 17.3 before 17.3.2.”*

However, there was no evidence of active exploitation of this vulnerability, GitLab recommended its users update their GitLab instances as soon as possible.

[The Hacker News](#) / [Bleeping computer](#)

## GitLab Status info: a total of 12 incidents

JULY '24

Incidents that had a critical impact – 2 incidents

**Total time of service disruption – 783 hours 9 minutes**

Incidents that had a major impact – 5 incidents

**Total time of partial service disruption – 74 hours 54 minutes**

Incidents that had a minor impact – 2 incidents

**Total time of degraded performance – 22 hours 53 minutes**

Operational issues – 3 incidents

**Total time of operational issues – 20 hours 8 minutes**

GitLab reported 12 incidents, including 2 critical ones totaling over 783 hours of disruption. The most significant issue lasted from June 6 to July 8, impacting bulk import functionality for nearly 779 hours. Another critical incident on July 11 caused a 4-hour global unavailability of core GitLab services for some users, affecting numerous components.

Also, there were five major partial disruptions totaling around 75, including prolonged issues with package manifests and merge requests. Minor performance degradation occurred in two incidents, lasting 22+ hours, primarily affecting repository and package access.

---

# Critical patch – security flaw allows unauthorized pipeline jobs

July brought the GitLab's release of updates for 6 detected security flaws, including a critical vulnerability CVE-2024-6385 with a CVSS score of 9.6 out of 10 possible.

The flaw highlights ongoing risks, as similar vulnerabilities in June and August with around the same critical severity score were patched.



*"We strongly recommend that all installations running a version affected by the issues described below are upgraded to the latest version as soon as possible," states [GitLab's advisory](#).*

[Bleeping Computer](#)

[The Hacker News](#)

[Security Week](#)

This bug, affecting GitLab CE/EE versions 15.8 to 17.1, allows attackers to execute pipeline jobs as arbitrary users under specific conditions.

## GitLab Status info: a total of 5 incidents

JUNE '24

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption – 52 hours 6 minutes**

Incidents that had a minor impact – 2 incidents

**Total time of degraded performance – 3 hours 45 minutes**

Operational issues – 1 incident (started in May)

**Total time of operational issues – 845 hours 36 min**

In June, GitLab experienced 5 incidents, including two major ones that led to partial service disruption that lasted in total for over 52 hours – more than a working week (over 6 days!).

---

# A hacker can access and run the GitLab CI/CD pipeline of any GitLab user due to a bug

GitLab has released security updates addressing 14 vulnerabilities, including a critical flaw, CVE-2024-5655, that could allow an attacker to run CI/CD pipelines as another user. We've already mentioned previously in this blog that it was the third similar vulnerability that GitLab faced in 2024.

The case was reported to GitLab through its HackerOne bug bounty program. This vulnerability, rated 9.6 out of a maximum of 10 on the CVSS scale, impacts versions 17.1 prior to 17.1.1, 17.0 prior to 17.0.3, and 15.8 prior to 16.11.5 in both the Community and Enterprise Editions.

[According to GitLab](#), the hosting service “we have not found evidence of abuse of this vulnerability on the platforms managed by GitLab, including *GitLab.com* and *GitLab Dedicated instances*.”

However, GitLab urgently advises its users to update their GitLab instances. The updates, available in versions 17.1.1, 17.0.3, and 16.11.5, introduce two breaking changes: GraphQL authentication using CI\_JOB\_TOKEN is now disabled by default, and pipelines no longer run automatically after a merge request's target branch is merged.

Additional high-severity flaws include CVE-2024-4901 (stored XSS vulnerability, CVSS 8.7), CVE-2024-4994 (CSRF in GraphQL API, CVSS 8.1), CVE-2024-6323 (authorization issue in global search, CVSS 7.5), and CVE-2024-2177 (cross-window forgery in OAuth authentication, CVSS 6.8). These vulnerabilities could lead to unauthorized actions, sensitive data leakage, and exploitation via malicious payloads.

## [The Hacker News](#)

### **GitLab Status info: a total of 3 incidents**

MAY '24

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption – 4 hours 19 min**

Incidents that had a minor impact – 1 incident

**Total time of degraded performance – 1 hour 7 minutes**

In May, GitLab reported 3 incidents including 2 issues of partial service disruptions totaling over 4 hours. These incidents impacted the API, website, and CI\CD services.

---

# Hackers can take over GitLab accounts exploiting high-severity vulnerability

A high-severity XSS vulnerability, tracked as CVE-2024-4835, in the VS Code editor (Web IDE) that could allow unauthenticated attackers to take over user accounts by tricking users into interacting with malicious pages was patched by GitLab developers. Due to the absence of authentication necessity, user interaction increases the attack's complexity.

This vulnerability, as well as other six medium-severity flaws, was addressed in versions 17.0.1, 16.11.3, and 16.10.6 of GitLab Community Edition (CE) and Enterprise Edition (EE).

Other important-to-mention medium-severity issues include a CSRF vulnerability via the Kubernetes Agent Server (CVE-2023-7045) and a denial-of-service bug (CVE-2024-2874) that could disrupt loading GitLab web resources.

GitLab **strongly recommended** upgrading to the described patched versions immediately as the updates aimed to enhance security and protect users from potential exploitation.

[Bleeping Computer](#)

---

# Severe GitLab password reset vulnerability is actively exploited by attackers

The U.S. Cybersecurity and Infrastructure Security Agency warned about GitLab's critical vulnerability, tracked as CVE-2023-7028, to its Known Exploited Vulnerabilities (KEV) catalog due to active exploitation in the wild.

The flaw, with the highest CVSS score possible of 10, was introduced in GitLab version 16.1.0, allows an attacker to take over GitLab users' account by sending password reset emails to unverified email addresses. Although users with two-factor authentication are less vulnerable to account takeover, their accounts can still be subjected to password resets.

By exploiting this vulnerability, attackers get the possibility to steal GitLab users' sensitive information, credentials, and malicious code insertion in source code repositories, potentially resulting in supply chain attacks. Moreover, attackers may embed malware in CI/CD pipelines to exfiltrate data or create backdoors, undermining system integrity and security.

The issue has been patched in GitLab versions 16.5.6, 16.6.4, 16.7.2, and earlier versions via backports. Federal agencies are mandated to have applied these fixes by May 22, 2024, to protect against ongoing exploitation.

[The Hacker News](#) / [SC Media](#)

## GitLab Status info: a total of 10 incidents

APRIL '24

Incidents that had a critical impact – 1 incident

**Total time of service disruption – 1 hour 30 minutes**

Incidents that had a major impact – 7 incidents

**Total time of partial service disruption – 50 hours 34 minutes**

Incidents that had a minor impact – 2 incidents

**Total time of degraded performance – 43 hours 59 minutes**

In April, GitLab reported 10 incidents, including one highly severe service disruption that lasted for over an hour. Also, there were 7 partial service disruptions that accounted for a total of 50+ hours, with issues such as stuck merge requests, SaaS runners failing, and errors managing Kubernetes agents.

The other 2 incidents caused degraded performance, totaling about 44 hours, with notable issues like errors in downstream pipelines and dependency proxy failures. The disruptions affected key components, including the website, API, CI/CD runners, and the customer portal, all hosted on Google Compute Engine.

# Hackers abuse GitLab to spread malware

In April, Bleeping Computer experts investigated and reported how threat actors can abuse GitHub comments to push malware. However, later after checking GitLab, it turned out that the service provider tends to the same issue – the git hosting service is susceptible to abuse of its comments feature, allowing threat actors to upload malicious files that appear to be hosted on legitimate repositories.

This exploit enables attackers to upload files, including malware, to GitLab's CDN under URLs resembling official project repositories. The links remain live even if the associated comment is never posted or is deleted, making them a potential tool for deception.

For example, files uploaded using this flaw can mimic releases from trusted open-source projects like

Inkscape and Wireshark, misleading users into downloading counterfeit software.

```
https://gitlab[.]com/inkscape/inkscape/uploads/edfdbc997689255568a7c81db3f3dc51/Inkscape-2024-Latest.exe  
https://gitlab[.]com/wireshark/wireshark/uploads/b4162053fbb4dc6ee4f673c532009e16/Wireshark-v4.2.4-stable-release.exe
```

Source: [Bleeping Computer](#)

Though GitLab requires its users to sign in before they upload the files, it can't prevent bad actors from uploading malicious files.



*"Since virtually every software company uses GitHub or GitLab, this flaw enable allow threat actors to develop extraordinary crafty and trustworthy lures," states Bleeping Computer.*

[Bleeping Computer](#) / [GridinSoft blog](#)

# JsOutProx malware attacks financial institutions in APAC and MENA abusing GitLab

In early April, Resecurity reported that a new version of JSOutProx, a sophisticated attack framework targeting financial organizations in the APAC and MENA regions, had been detected.

JSOutProx uses JavaScript and .NET, leveraging .NET (de)serialization to interact with JavaScript modules on victims' systems and load plugins for further malicious activities. Initially linked to SOLAR SPIDER's phishing campaigns since 2019, the malware has been used in attacks involving fake SWIFT and Moneygram notifications to target banking customers.

In 2024 a few incidents involving a system integrator in Saudi Arabia and financial customers targeted through impersonation schemes using the *"mike.will@my[.]com"* took place.

GitLab? On March 27th, 2024 Resecurity researchers noticed a new malware sample on GitLab (before the attackers used GitHub in their malicious schemes):



*"The actor registered multiple accounts on GitLab around March 25, 2024, and used them to deploy repositories containing malicious payloads," is stated in the Resecurity Report.*



Source: [Resecurity blog](#)

schemes – prove that JSOutProx is still an evolving and significant threat, especially when it comes to financial institutions.

[Cyber Security News](#)  
[Security Affairs](#)

After the successful delivery of the malicious code, the thread actor removes the GitLab repo and creates an absolutely new one. It may be related to the fact that the hacker exploits multiple malicious payloads and adapts to its target.

These two facts – the discovery of the new version of JSOutProx and GitLab's abuse it the malicious

## GitLab Status info: 6 incidents

MARCH '24

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption – 66 hours 57 hours**

Incidents that had a minor impact – 4 incidents

**Total time of degraded performance – 41 hours 19 minutes**

In March, GitLab communicated in its GitLab Status 6 incidents. There were 2 major issues that lasted a total of around 67 hours and 4 minor incidents that caused degraded performance for GitLab users for over 41 hours. Affected components included the website, API, CI/CD runners, and background processing, with services hosted on Google Compute Engine, AWS, and Zendesk.

---

# GitLab security flaw allows threat actors to inject malicious scripts

In its [March patch release](#), GitLab addressed critical vulnerabilities that could permit attackers to inject malicious scripts and cause the DoS attack.

Discovered in GitLab Community Edition and Enterprise Edition, a Stored Cross-Site Scripting (XSS) vulnerability, – the CVE-2023-6371 security flaw with a severity of 8.7 – could allow the attacker to perform arbitrary actions on behalf of the affected users by injecting a crafted payload into a wiki page. The vulnerability affected all versions of GitLab CE/EE prior to 16.8.5, from 16.9 before 16.9.3, and from 16.10 to 16.10.1.

Another security flaw that GitLab had to address in March was DoS using crafted emojis. Tracked as CVE-2024-2818 with the CVSS of 4.3, the flaw could permit the threat actors to trigger a denial of service by using the maliciously crafted description parameter for labels.

To improve GitLab users' security posture, GitLab strongly advised them to update the affected versions.

[Cyber Security News](#)

## GitLab Status info: 6 incidents

FEBRUARY '24

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption: 1 hour 54 minutes**

Incidents that had a minor impact – 3 incidents

**Total time of degraded performance – 18 hours 6 minutes**

Planned Maintenance – 1 incident

**Total time of planned maintenance – 1 hour**

GitLab experienced 6 incidents of different impacts that might influence GitLab users in February. Those events included 2 major ones causing about 2 hours of partial service disruption, both involving intermittent job failures with GitLab SaaS runners, and 3 minor incidents which led to 18+ hours of degraded performance, primarily affecting the API and website components.

---

## Discovered exposed API secrets could impact major tech tokens

The security research team at Escape discovered over 18K vulnerable API secrets after scanning 189M+ URLs. Of the disclosed secrets, 41% were highly critical, which means that they could lead to financial risks for companies. Among the exposed secrets were hundreds of GitLab and GitHub tokens, Stripe, RSA private keys, OpenAI keys, AWS tokens, Slack and Discord webhooks, etc.

To mitigate the risks, the Escape researchers advise centralizing token management and rotating tokens regularly. Moreover, it's possible to monitor token usage patterns, restrict token scope, allocate tokens to specific teams or services, and assign appropriate permissions to team members.

[Help Net Security](#)

## GitLab Status info: 4 incidents

JANUARY '24

Incidents that had a major impact – 2 incidents

**Total time of partial service disruption – 13 hours 12 minutes**

Incidents that had a minor impact – 2 incidents

**Total time of degraded performance – 223 hours 46 minutes**

In January, GitLab users might face 2 major incidents causing over 13 hours of partial service disruption that GitLab experienced.

Moreover, there were 2 minor incidents that resulted in 223+ hours of degraded performance.

---

## Possible file overwriting due to workspace creation flaw

On January 25th, 2024 GitLab released the fixes to address a security flaw, tracked as CVE-20240-402 with a CVSS score of 9.9, in its Community Edition and Enterprise Edition.

Using the vulnerability, bad actors could write files to arbitrary locations while creating a workspace.



*“An issue has been discovered in GitLab CE/EE affecting all versions from 16.0 prior to 16.5.8, 16.6 prior to 16.6.6, 16.7 prior to 16.7.4, and 16.8 prior to 16.8.1 which allows an authenticated user to write files to arbitrary locations on the GitLab server while creating a workspace. This is a critical severity issue (CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H, 9.9).”*

[GitLab Critical Security Release](#)

**Also, in the January patch release, GitLab addressed a few other security flaws of medium severity:**

- ReDoS in Cargo.toml blob viewer, tracked as CVE-2023-6159, with a CVSS of 6.5, which allowed an attacker to trigger a Regular Expression Denial of Service via a Cargo.toml containing maliciously crafted input.
- Arbitrary API PUT requests via HTML injection in user's name, tracked as CVE-2023-5933, with a CVSS of 6.4.

- Disclosure of public email in Tags RSS Feed, tracked as CVE-2023-5612 with the CVSS of 5.3, which allowed to read the user email address via tags feed despite the fact that the visibility in the user profile was disabled.
- Non-member update of MR Assignees of owned MRs, tracked as CVE-2024-0456 with the CVSS of 4.3, which allowed an unauthorized threat actor to assign arbitrary users to MRs that they created within the project.

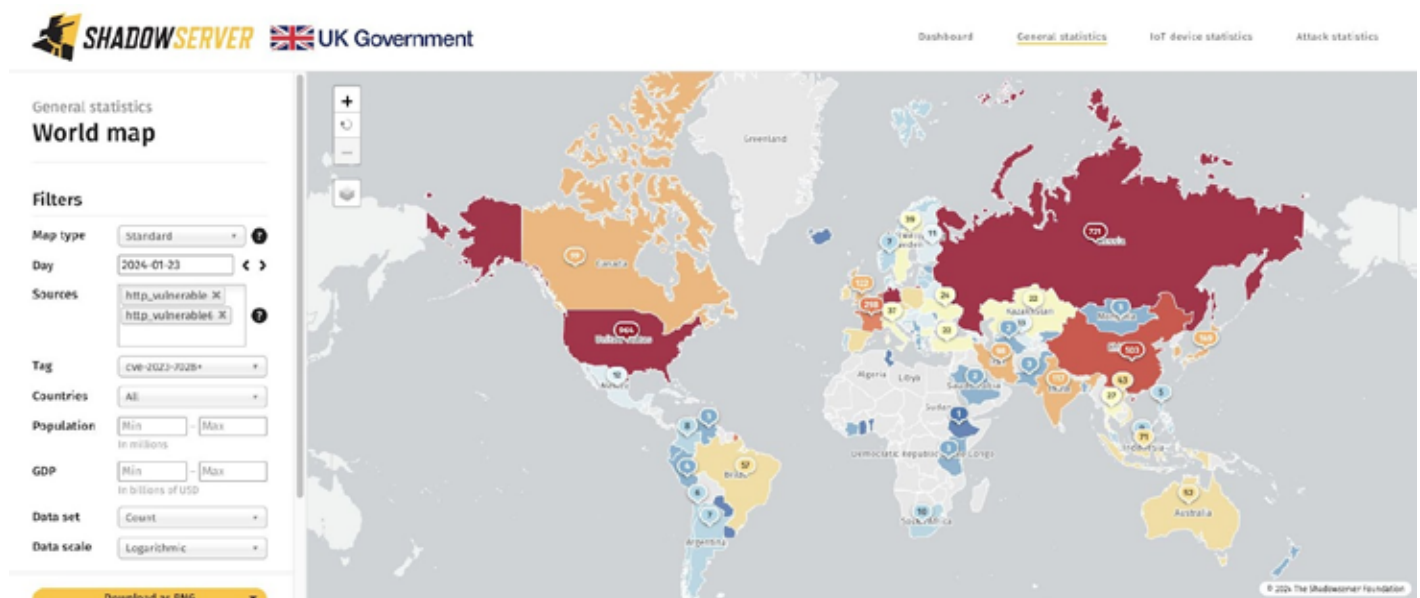
To mitigate the issues, GitLab strongly recommends its users who are running an affected version to upgrade their instance to the latest version as soon as possible.

[The Hacker News](#) / [Help Net Security](#)

# More than 5K servers are vulnerable to assaults that use zero-clickaccount takeover

Due to the critical flaw CVE-2023-1028, with a CVSS score of 10 of a maximum of 10!, over 5,379 internet-exposed GitLab instances were reported to be vulnerable to a zero-click account takeover flaw. Using the vulnerability, the threat actors can change the user's password to GitLab and take control of the account by sending password reset emails under the attackers' control.

According to Shadowserver reports, the most vulnerable GitLab servers were found in the USA with 964 ones, and Germany with 721 ones.



Source: [Shadowserver reports \(X\)](#)

---

On January 11th, 2024, 13 days prior, GitLab released patches for the impacted GitLab Community and Enterprise Edition versions 16.1 (prior to 15.1.5), 16.2 (prior to 16.2.8), 16.3 (prior to 16.3.6), 16.4 (prior to 16.4.4), 16.5 (prior to 16.5.6), 16.6 (prior to 16.6.4), and 16.7 (prior to 16.7.2).

Later GitLab advised its users to check where they are vulnerable to the issue, and if so, rotate all their credentials, API tokens, certificates, and other secrets. Moreover, it's worth enabling 2FA on all the accounts together with security updates.

[Bleeping Computer](#) / [The Register](#)

# Takeaway

Comparing year to year, it's possible to confirm that cyberattacks are growing and hackers are eager to use any possible vulnerability or security flaw to access their victims' critical data. Attackers try to do their best to target GitLab as it is a hub for hosting various types of different organizations' sensitive corporate data, such as API keys, or proprietary code, which open the doors for threat actors to further malicious actions. Among those, there can be compromising the breached GitLab user's repositories, supply chain attacks, etc.

GitLab is a secure hosting service that instantly reacts to incidents... and in 2024, the service provider experienced over 90 incidents, including those that might lead to service disruption, totaling over



798+ hours (around 99 working days!), and partial disruption events – 44 incidents in 2024 with the summed-up time of over 585 hours of upset. Moreover, GitLab always takes immediate actions to mitigate security flaws. However, there is still a shared responsibility model between the service provider and a GitLab user. It states that both GitLab and its users are responsible

---

for the security of their data. From its side, GitLab does everything to help its users save their data, but let's not forget that GitLab account data safety is the user's duty. Thus, it's important to implement security best practices, including updating GitLab instances should the GitLab's

 Azure DevOps 2024 report

# The State of Azure DevOps Threat Landscape - 2024 In Review

1

incident with unhealthy status

~2<sub>hrs</sub>

of disruption



GitProtect  
by Xopero ONE

# The State of Azure DevOps Threat Landscape – 2024 In Review

The year 2024 is over, so it's time to sum up what threats were the most dangerous for DevOps and PMs. Outages, degraded service performance, vulnerabilities, cyberattacks, ransomware – all of those were appearing in media headlines all year round. Thus, for the third year in a row, we've decided to analyze incidents related to Git hosting services, like Azure DevOps, GitHub, GitLab, and Atlassian.

Our first article in a DevOps threat landscape series is dedicated to Azure DevOps. Let's dive into the news, reports, and Azure DevOps status to see which threats Azure DevOps users had to deal with in the previous year.

Azure DevOps recorded incidents during the year



**Azure DevOps Status: 3 incidents**

DECEMBER '24

Incidents that had a degraded impact – 3 incidents

**Total time of incidents with degraded performance – 39 hours 21 minutes**

In December some Azure DevOps users in Europe and Brazil experienced 3 incidents with degraded impact, totaling almost 40 hours.

Those 3 cases of degradation performance were impacting Azure DevOps services, including Boards, Repos, Pipelines, Test Plans, Artifacts, and other services.

**Azure DevOps Status: 12 incidents**

NOVEMBER '24

Incidents that had a degraded impact – 12 incidents

**Total time of incidents with degraded performance – 19 hours 55 minutes**

Azure DevOps experienced 12 incidents in November. All of the incidents Microsoft classified as degraded impact totaled around 20 hours of disruption.

The most significant incident occurred on November 18th, impacting North European users for over 6 hours, who had difficulties accessing Boards, Repos, Pipelines, and Test Plans.

Other incidents were shorter, targeting different regions, including Asia Pacific, the United States, Australia, and Brazil.

Among the services that were impacted in November were pipelines, boards, repos, and other core services.

### **Azure DevOps Status: 7 incidents**

OCTOBER '24

Incidents that had a degraded impact – 7 incidents

**Total time of incidents with degraded performance – 42 hours 55 minutes**

In October Azure DevOps users globally experienced 7 degraded incidents with a total time for disruptions of over 40 hours.

The incidents included brief availability degradations across test plans, boards, repos, pipelines, and other services.

### **Azure DevOps Status: 6 incidents**

SEPTEMBER '24

Incidents that had a degraded impact – 6 incidents

**Total time of incidents with degraded performance – 168 hours 34 min**

In September Azure DevOps faced 6 degraded incidents, totaling around 168 hours of disruption... almost 7 days! The most prolonged incidents included a 93-hour 28-minute issue that affected connections in Pipelines.

Other smaller incidents impacted Artifacts and other Azure DevOps services.

### **Azure DevOps Status: 5 incidents**

AUGUST '24

Incidents that had a major impact – 1 incident

**Total time of incidents with unhealthy performance – 1 hour 21 minutes**

Incidents that had a degraded impact – 4 incidents

**Total time of incidents with degraded performance – 11 hours 8 minutes**

Azure DevOps users could experience 5 incidents of different severity in August. The total time of disruption compounds 12 hours and 29 minutes.

The major incident, lasting 1 hour and 21 minutes on August 5, caused intermittent errors across core services like Boards, Repos, and Pipelines in Brazil and the United States.


# Services across North and Latin America are taken down due to Microsoft Azure outage



Multiple Azure customers across North and Latin America were taken down due to a Microsoft Azure outage. The company said that the incident impacted services that leverage Azure Front Door (AFD).


As Microsoft stated, the incident happened due to *“a recent configuration change made by an internal service team that uses AFD. Once that change was understood as the trigger event, we initiated a rollback of the configuration change which fully mitigated all customer impact.”*


## Intermittent errors connecting to Azure DevOps


### Event log



**Final update**

 Birendra Acharya, 05/08/2024, 22:16  
 The issue is now fully mitigated.  
  
 This issue was caused by an AFD outage which impacted other Microsoft services as well. <https://azure.status.microsoft.com/en-us/status/history/> , tracking Id: 0N\_5\_PQ0.  
  
 Our engineers will be investigating this further to learn from and reduce the risk of potential recurrences. We apologize for the impact this had on our customers.


**Update**

 Birendra Acharya, 05/08/2024, 21:35  
 We are investigating an outage across all Azure DevOps services. The incident is intermittently preventing users from connecting to Azure DevOps.










**Initial communication**

 Azure DevOps Incident Scout, 05/08/2024, 20:43  
 Our engineers are currently investigating an event impacting Azure DevOps. The event is being triaged and we will post an update as soon as we know more.

### Event information

State	Start	End
Resolved	05/08/2024, 20:20	05/08/2024, 21:41

### Impact information

Severity	Services	Geographies
 Unhealthy	 Core services  Boards  Repos  Pipelines  Test Plans	 Brazil  United States

Source: [Azure DevOps Status](#)

As a result, for around 2 hours Azure DevOps users reported errors connecting to Azure services and logging problems.

This outage followed another massive incident, an Azure outage, that lasted for 9 hours and impacted many Azure and Microsoft 365 services. The reason for that 9-hour outage was a volumetric TCP SYN flood DDoS attack which targeted many Azure Front Door and CDN sites.

[Bleeping Computer](#) / [DCD](#)

#### **Azure DevOps Status: 22 incidents**

JULY '24

Incidents that had a degraded impact – 22 incidents

**Total time of incidents with degraded performance – 376 hours 08 minutes**

In July, Azure DevOps users around the world saw 22 incidents, all with degraded impact, resulting in a total disruption time of more than 200 hours (more than 8 days!).

The key incidents included an over 275-hour-long disruption affecting test case order changes across multiple regions and a 38-hour availability degradation in Service Hooks in SEA.

Other incidents were affecting core Azure DevOps services like Boards, Pipelines, Test Plans, and Repos across various regions, including Europe, South and North America.

# Storage incident in the Central US impacted Azure DevOps users all over the world

On July 18th Azure DevOps users might experience issues accessing their Azure DevOps organization.

In their [post-mortem](#), Microsoft explained that *“This outage lasted roughly from 21:40 to 02:55 UTC. During this outage, customers may have experienced a range of impact from a full outage of their organization to partial degradation with certain functionality unavailable such as Artifacts, Packaging, Test, etc.”*

The outage was caused by VM availability in the Central US region and led to all Azure DevOps organizations located in that region being inaccessible. For over an hour, Azure DevOps users were receiving a 503 “Service unavailable” message when they were trying to access their Azure DevOps instances. Moreover, around 1.4% of organizations outside of the Central US region might also be experiencing some kind of performance degradation at that time as well.



Source: Bleeping Computer – SENSEI character sheet from leaked documents

## Azure DevOps Status: 15 incidents

JUNE '24

Incidents that had a degraded impact – 15 incidents

**Total time of incidents with degraded performance – 21 hours 30 minutes**

The incidents that Azure DevOps users might face in June were all of degraded severity with the total time of disruption of 21+ hours.

The most significant incident was a 7-hour 26-minute service connection degradation affecting Pipelines.

Other incidents were much shorter and impacted such Azure DevOps services as Boards, Pipelines, Repos, and Test Plans.

---

# Threat actors might gain unauthorized access to Cloud Azure resources

Ten Azure services, including Azure DevOps, Azure, Azure Load Testing, Azure API Management Application Insights, etc. were found vulnerable to a bug that allowed an attacker to bypass firewall rules and gain unauthorized access to cloud resources by abusing Azure Service Tags.

A researcher from the cybersecurity firm Tenable [stated](#):

*“This vulnerability enables an attacker to control server-side requests, thus impersonating trusted Azure services. This enables the attacker to bypass network controls based on Service Tags, which are often used to prevent public access to Azure customers’ internal assets, data, and services.”*

While there’s no evidence of this vulnerability being exploited in the wild, Microsoft advises customers to review their use of Service Tags, strengthen validation controls, and implement robust authentication to ensure only trusted traffic is allowed.

In its guidance issued on June 3rd, 2024, [Microsoft states](#) that “Service tags are not to be treated as a security boundary and should only be used as a routing mechanism in conjunction with validation. [...] Service tags are not a comprehensive way to secure traffic to a customer’s origin and do not replace input validation to prevent vulnerabilities that may be associated with web requests.”

[The Hacker News](#) / [Bleeping Computer](#)

### Azure DevOps Status: 10 incidents

MAY '24

Incidents that had a degraded impact – 10 incidents

**Total time of incidents with degraded performance – 18 hours 38 minutes**

In May, Azure DevOps encountered 10 incidents with a degraded impact, amounting to a total disruption time of over 18 hours.

Users from all geographies might experience disruptions while working with Boards, Repos Pipelines, and other Azure DevOps services. The longest incident that month, lasting 6 hours and 56 minutes, involved failures in Pipelines due to Workload Identity federation-based ARM service connections.

### Azure DevOps Status: 12 incidents

APRIL '24

Incidents that had a degraded impact – 12 incidents

**Total time of incidents with degraded severity – 42 hours 25 min**

Users of Azure DevOps in different locations might experience 12 incidents with degraded impact, varying from a few minutes to hours. Summing up, Azure DevOps experienced more than 42 hours of disruptions in April.

The longest incident lasted over 19 hours. During that time Azure DevOps users all around the world might experience slowness and disconnections from the service, impacting Boards, Repos, Pipelines, Artifacts, etc.

# Failures and slowness across Azure DevOps in all geographies

On April 25th, Azure DevOps users all around the world might experience service availability degradation – the slowness and intermittent failures across the service. The issue stemmed from a severe performance regression in a frequently used stored procedure within the Token Service database, increasing execution time by nearly 5000x.



Source: [Azure DevOps post-mortem](#): Global request failure rate %

As Microsoft stated in its [post-mortem](#) later *“The incident was caused by a significant regression in the performance of a stored procedure with a high frequency of calls in the Token Service.”*

Among the affected Azure DevOps experiences were Azure Pipelines with an average delay of over 3 minutes, Personal Access Token (PAT) usage, and increased failure rates for Azure DevOps extensions, with the global request failure rate peaking at around 20%.

# Reduced availability of Azure DevOps services across the globe

Some Azure DevOps users were experiencing network-related errors, unexpected cancellations, and slowness while they were trying to access [large files](#) over HTTPS for a few days (April, 22nd – April 24th). That led to reduced availability of Azure repositories, boards, and pipelines.

As Microsoft explained in its [post-mortem](#), the reason turned out to be an update that the service provider rolled out sometime prior:

*“As part of ongoing work to make Azure Front Door more reliable, the team rolled out an update, which included a specific change for how packets are processed. Due to this change, some traffic was not processed correctly, which triggered clients to resend the packets, and this led to higher latency. As such, this issue primarily affected scenarios with large downloads and responses.”*

Azure DevOps engineers quickly detected and addressed the issue.

## Azure DevOps Status: 7 incidents

MARCH '24

Incidents that had a degraded impact – 7 incidents

**Total time of incidents with degraded performance – 36 hours 22 min**

**Azure DevOps Status: 15 incidents**

JUNE '24

Incidents that had a degraded impact – 15 incidents

**Total time of incidents with degraded performance – 21 hours 30 minutes**

In March 2024, Azure DevOps experienced seven incidents resulting in degraded service across various regions, totaling over 36 hours of disruption.

The most significant issue happened on March 27–28, when Analytics dashboard widgets failed to load globally for over 22 hours.

Other incidents included availability degradations in Central US, West Europe, and the UK, affecting Boards, Repos, Pipelines, and Test Plans. The artifacts were also impacted twice, showing degraded performance in West Europe and the Central US region.

**Azure DevOps Status: 12 incidents**

FEBRUARY '24

Incidents that had a degraded impact – 12 incidents

**Total time of incidents with degraded performance – 48 hours 45 min**

---

Azure DevOps experienced 12 incidents which led to over 48 hours of degraded service across multiple regions in February 2024.

The most prolonged disruption lasted over 26 hours, affecting extensions in the United States.

Europe saw the highest concentration of issues, with multiple availability degradations impacting Boards, Repos, Pipelines, and Artifacts, particularly in West Europe.

Other incidents included pipeline delays in the UK, a global issue preventing Pipeline Artifact downloads, and deprecated pipeline tasks becoming unusable.

---

# Critical Azure DevOps flaw could allow remote code execution

Microsoft addressed 73 security vulnerabilities and 2 actively exploited zero-days in February. Among the patches for different Microsoft products, there was one for Azure DevOps Server to address the CVE-2024-20667 security flaw tagged as “important” within the severity level.

Using the vulnerability, an attacker could run unauthorized code on the user’s Azure DevOps server and potentially compromise sensitive data or disrupt the user’s services. However, there should be specific conditions for an attacker to succeed in an attack within this vulnerability:

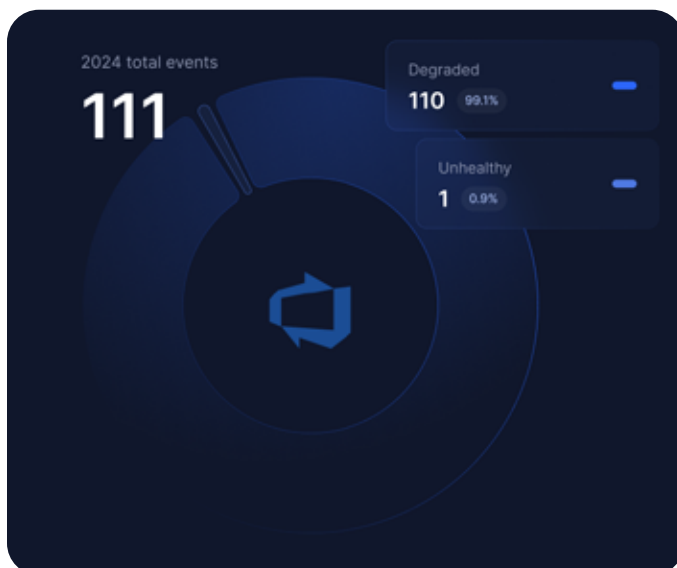
*“Successful exploitation of this vulnerability requires the attacker to have Queue Build permissions and for the target Azure DevOps pipeline to meet certain conditions for an attacker to exploit this vulnerability,”* is stated in Microsoft’s security updates.

The February updates are essential for users to protect their Azure DevOps Server against this remote code execution bug.

[Bleeping computer](#)

# Takeaway

After analyzing all the events that impacted Azure DevOps and thoroughly going through the [Azure DevOps Status](#) (here we managed to analyze only the period from February 2024 to December 2024), we counted that the service experienced 111 incidents of degraded performance, which gives us the total time of disruption – 826 hours and 02 minutes. Thus, Azure DevOps experienced some kind of disruption for about 103 working days, including an outage that took place in Brazil and the United States.



Let's not forget that Azure DevOps is a secure platform and Microsoft takes all the necessary measures to make its services available. It patches vulnerabilities and provides updates to Azure DevOps users to know how to deal with them. Here we shouldn't forget about the Shared Responsibility Model that Microsoft follows. Within it, Microsoft is responsible for its services staying available while users should take care of their data and its security, availability, and recoverability.

To have peace of mind that their source code and project data are secured, Azure DevOps users should implement [security best practices](#), which include MFA, monitoring, backup, etc.

---

Moreover, in view of meeting not only security but also compliance requirements, Azure DevOps users should consider backing up their DevOps and PM's environment.

[GitProtect backup and DR software for Azure DevOps](#) helps admins ensure that their company's critical data is accessible and recoverable in any event of disruption. The solution helps organizations to follow [backup best practices](#) that leave no place for data loss.

---

# DevOps Backup Best practices

No matter which DevOps tool you use, GitHub, GitLab, Azure DevOps, or Atlassian's Bitbucket and Jira, you need to be sure that your Intellectual property, hours of work, reputation, and custom trust are steadfast.

**For that, you should ensure that your backup option covers three main aspects of backup:**

- **Backup Performance** – to make you easily operate with your backups,
- **Backup Security** – to provide the most secure features to protect your data,
- **Restore and Disaster Recovery** – to guarantee fast restore under any disaster scenario.

---

# Backup Performance

## Full data coverage

To be comprehensive, your backup should include as much information as possible. Thus, whether you back up GitHub, Bitbucket, GitLab, Azure DevOps, or Jira, your backup plan should have full data coverage in any deployment model.

Thus, when it comes to git hosting services, your backup plan should cover all repositories and their related metadata, including wiki, issues, LFS, webhooks, pull requests, issue comments, etc. In turn, backup for Jira Software and Jira Service Management should include all project management and communication data, such as Jira Assets, Jira Actions, projects, Jira issues and comments, audit logs, notifications, attachments, workflows, boards, versions, and more.

Moreover, you should be able to set up different custom backup plans. It will help you adjust your data protection policy to the requirements, workflow, and structure of your organization.

The best practice is to have a few backup plans: one for unused repositories that you have to keep for future reference or compliance needs, and another for critical DevOps data that change daily or even more frequently, such as when using the suggested GFS rotation scheme, or Forever Incremental.

## Save your storage – incremental and differential backups

To minimize backup storage space, optimize backup performance, and control bandwidth, your backup solution should include only changed data from the previous copy. Ideally, you should have the possibility to specify various retention and performance plans for each kind of copy (full, incremental, and differential).

## Different deployment models – SaaS and On-Premise

You should always have a choice... Thus, it's vital that your backup software can be installed and run on both the cloud and/or your own infrastructure.

With a SaaS model, you may install it without having to set aside any extra hardware for a local server because the service is hosted on the cloud infrastructure of the provider. Its management and upkeep are taken care of for you, and the service provider ensures that it will continue to run.

When you deploy software on-premises, it is installed on a system that you own and manage, allowing it to function locally in your environment. The ability to install it on any computer – Windows, Linux, macOS, and even well-known NAS devices – is a great bonus. As the copies will be used within the local network, you won't have to worry about network connectivity problems. What's more, this deployment model will do backups quicker and more effectively as everything is run within your own infrastructure.

What's important is that the deployment model of your choice should be independent of your data storage compatibility. For example, with GitProtect.io, you can both get the GitProtect unlimited cloud storage (which is always included in the license) and assign as many storage destinations as you need, whether they're local or cloud. The backup provider supports AWS, Azure Blob Storage, Wasabi, Google Cloud Storage, Backblaze B2, or any other compatible with S3 cloud storage, on-premise storage instances (CIFS, NFS, SMB network shares, local disk resources), and hybrid and multi-cloud environments.

## The 3-2-1 backup rule – a robust protection of your DevOps data

You should be able to create an infinite number of on-premises or cloud (preferably both) storage instances with your DevOps backup software. It will allow you to follow the [3-2-1 backup rule](#) or other modern backup strategies (the 4-3-2 or the 3-2-1-1 backup rules), duplicate backups across storage instances, and eliminate any outage or event of a disaster.

GitProtect.io is a multi-storage system. It means that you can assign as many storage destinations for your DevOps backups as you need within your organization, legal, or compliance requirements. **Thus, you can keep your copies:**

- in the cloud (GitProtect Cloud Storage, Azure Blob Storage, AWS, Blackblaze B2, Google Cloud Storage, Wasabi, or any other public cloud S3-compatible,
- on-premise (SMB, CIFS, NFS network shares, local disk resources),
- multi-cloud or hybrid environments.

Once you decide to back up your DevOps tools with GitProtect.io, you will get an unlimited GitProtect Cloud Storage for free. So, you can start protecting your DevOps data immediately.

## Use Case

Let's imagine that within your organization and compliance needs you should keep your data on [your Cloud Storage]. However, because of your consciousness, you decide to send your backup copies to your local device.

Suddenly, one day, your git hosting service is down. What's worse, [your Cloud Storage] experiences an outage, as well, and your organization's workflow is paralyzed. Yet, don't forget that you made some backup copies to your local device, so you can instantly recover your data with a click.

All you need to do is log in to your GitProtect.io account, pick up the backup plan assigned to your local device, choose the last copy (it depends on the frequency of backups you set up), and recover the data. With the backup solution, you have different restore options – restore to the same git hosting provider's account, to your local machine, or cross-overly to another git hosting platform (e.g. from GitHub to GitLab or Bitbucket).

**Real life story:** IowaComputerGurus, a US leading supplier of custom application solutions based on Microsoft.NET Technology stack that provides exceptional design, development, website, and performance optimization services, uses GitProtect.io's multi-storage opportunities to ensure the company's GitHub data protection. Read the full story – IowaComputerGurus



*With source code being our most critical asset, knowing that it is protected and in a verified additional location gives us great comfort, even if we hope we never need to go back and retrieve it.”*

*Mitchel Sellers, CEO/Director of Development at ICG*

## Backup replication among storage instances

Backup replication is one of the most crucial factors you should keep in mind when deciding on backup software. Replication between storage instances helps to adhere to the 3-2-1 backup rule, as it provides you with the possibility to maintain consistent copies in several locations, ensuring redundancy and business continuity. Moreover, your backup software should permit you to replicate your DevOps data from any to any data storage, including cloud to cloud, cloud to local storage, and locally without any limitations.

GitProtect.io, for example, allows you to set up a replication plan within the menu of the central management console. To get started, you should simply specify the source and target storage, agent, and basic schedule.

## Flexible and unlimited replication

Replication can become one of the most critical features when choosing a backup provider. The thing is that most SaaS providers provide a retention period for your data only from 30 up to 365 days by default. However, it's not enough. Depending on the industry, the type of data you store in your repositories, how long you have to keep them, and when the data should be restored in the event of a failure, some businesses may need to retain some types of data for years.

**Thus, you should be able to define retention within:**

- the number of copies you need to keep,
- the time of each copy – how long your copy should be kept in storage,
- keep copies infinitely for archive purposes.

### **Important**

You should have the possibility to set those replication parameters separately for the full, differential, and incremental backups.

## Monitoring Center – the DevOps security in your hands

Even if you're not in charge of managing the backup software directly, you still can be responsible for keeping an eye on the backup performance, status updates, or maybe checking the one whose duty is to making changes to the settings that can affect your administrators. To put it simply, you need to have a monitoring center that is complex and tailored.

Custom [email notifications](#) and alerts are among the simplest methods to remain up-to-date without even logging in.

### That's why you should be able to set up:

- backup plan summary details, including successfully finished tasks, ones finished with warnings, failed, canceled, and not started tasks,
- recipients, so that you don't even need to have an account within your backup software infrastructure to stay informed about backup statuses,
- restore verification summary to have a full picture of your restore operations when you need them,
- storage capacity notification,
- status report of your plan,
- SLA report for compliance needs.

In the perfect scenario, for your DevSecOps team's convenience, you should have backup notifications sent to the software your team uses on an everyday basis, like Slack which is a staple for DevOps to collaborate and communicate. With [Slack notifications](#), you won't miss any important information regarding your backup.

---

Also, it should be possible for you to view the status of tasks in progress and historical events. If your backup software provides you with the **tasks section**, you will always have an understanding of actions in progress along with comprehensive information at hand.

Moreover, you need to have access to **advanced audit** logs through your GitHub, GitLab, Bitbucket, or Jira backup software. All of the information on the operation of services, software applications, backups, and data restore is usually kept in logs. Additionally, audit logs permit you to see which actions your admins perform. That can help to prevent some intentional malicious activity from their side if a case like that arises.

Another way to make monitoring simpler and less laborious for your development and operations teams' side is enabling the possibility of connecting those audit logs via **webhooks and API** to your external monitoring system and remote management software, like PRTG.

What's more, you should make sure that all the mentioned information – backup and restore management, monitoring, and other system settings are easily available from a single **central management console**. There you should be empowered with powerful visual statistics, data-driven dashboards, real-time actions, SLA auditing, and Compliance reports.

## The best way to bypass throttling – a dedicated GitHub/Bitbucket/Azure DevOps/GitLab account

When it comes to large corporate customers, it's recommended to create a dedicated GitHub, Bitbucket, Azure DevOps, or GitLab user account that you can connect to your Git backup software. This account you should use only for backup purposes.

### In this case, you will catch up with two issues:

- **boost your security**, as the user of this dedicated account will have access only to repos it needs to protect,
- **bypass throttling**, as each GitHub, Bitbucket, Azure DevOps, or GitLab user has their own pool of requests to the Git hosting API. It means that all the applications linked to this account operate within the same number of requests. As a result, your team can avoid these restrictions and carry out backup operations without any delays or limits thanks to the separate user.

However, if your organization is too big and possesses numerous repositories, you should think about creating a few dedicated users for backup purposes on your Git hosting service. Thus, once the first one exhausts the number of API requests, you won't need to attach another one – it will be done automatically, and the same procedure will happen to the next one. What is the result? Even the largest DevOps environment continues to function without interruption during the backup performance. information regarding your backup.

# Backup Security

## DevOps backup solution for SOC 2 and ISO 27001 compliance

Security is shown to be a top priority for the majority of businesses. Let's not forget that source code is the most valuable asset for any IT-related business. That's the reason your repository and metadata backup have numerous security features that can assist you in following the Shared Responsibility Model, strengthening your company's security posture, and guaranteeing data accessibility and recoverability. So to say, your DevOps backup software should enable your organization to maintain regulatory compliance while empowering your development and operations teams, as well as security teams.

You should make sure that your software provider and Data Center where your service is hosted follow international security standards, audits, and certifications like SOC 2, ISO 27001, FISMA, HIPAA, GDPR, etc.

### Here are the issues your backup software should successfully meet:

- AES encryption with the possibility to create a personalized encryption key,
- in-flight and at rest encryption,
- the opportunity to set the needed level of retention – flexible, long-term, and unlimited retention,
- the potential to archive old, unused repositories according to your legal and organizational requirements,
- all-in-one central monitoring and management,
- multi-tenancy and privilege-based access control,
- strict [security and legal measures](#) for the Data Center,
- ransomware protection for 360 cyber resilience and compliance,
- [Disaster Recovery](#) under any failure event and instant restoration.

## In-flight and at-rest AES encryption

There is no way to talk about data protection without an appropriate and reliable encryption level. You should encrypt your data at every stage: while and before it's on your device, during the data transmission, and finally at rest in your repository. Moreover, you should make sure that your software is encrypted with the Advanced Encryption Standard (AES). Since AES, which uses symmetric keys, the same key is used for both data encryption and decryption, is considered an unbreakable one.

**Ideally, you should be able to set an encryption level and strength for your data, choosing any of:**

- **low** – the one that requires a 128-bit encryption key (Cyber-Block Chaining mode) of the AES algorithm,
- **medium** – the one that operates in the same CBC mode as the low, but with a longer encryption key (192 bits),
- **high** – operated within the same CBC mode as low and medium but more secure encryption key – 256 bits.

It's important to note that backup time depends on the selected method of encryption. Also, it makes the load on the end device or selected features limited. That's why you should have a choice between different levels of encryption, all of which are still unbreakable.

---

Having your personal encryption key is essential to robust encryption. To safeguard user data, the majority of providers simply generate encryption keys. Yet GitProtect goes one step further by letting you customize your unique encryption key. You can provide a string of characters based on which your personal encryption key is generated. What's more, you will be able to provide us with your key only during the backup process using your own Vault. That will give you more control over your credentials and access and will enhance your data protection security.

## Zero-knowledge encryption protocol

Have you heard that it's crucial for your device to be unaware of the encryption key? It should get it only when the backup process takes place. You are the only one who knows the key in this case, even your device has no information about it.

Security personnel usually call this method a **zero-knowledge encryption**. Thus, when you pick up a backup solution for your DevOps data, ensure that it provides all AES data encryption methods, allows you to create your personal encryption key, and enables a zero-knowledge approach.

## Data Center – choose your preferred region

If you're security-conscious and build your business on solid secure principles, you understand how critical it is that to have a full picture of your data stored and managed. Your backup software provider's Data Center location may affect uptime, coverage, and application availability.

That's why choosing the most appropriate location to store your data is vital. You have this option right away with GitProtect.io; upon signup, you will need to choose whether to keep your management service in a Data Center located in the US (Washington DC.), the EU (Amsterdam), or the APAC region (Australia).

What's important is that the Data Center you choose adheres to the stringent security regulations and is certified within ISO 27001, EN 50600, SOC 2 Type II, EN 1047-2 standard, FISMA, SOC 3, DOD, HIPAA, DCID, PCI\_DSS Level 1 and PCI DSS, LEED Gold Certified, SSAE 16, and LEED.

Other things that should catch your attention are physical security measures, fire safety and suppression, regular frequent audits, and round-the-clock network and technological support.

## **Privileged access management monitoring – how to share the duties**

Employee accountability sharing can improve your team's morale, don't you agree? Moreover, it can help your team to speed up operations and have a better view of the business. That's why, with your DevOps backup, you should be able to add new accounts, set privileges and roles to assign and delegate responsibilities to other members of your team and administrators. All of that will allow you to have more control over data protection and access controls.

The central management and monitoring console is the answer to this issue. With it, you can have audit logs with the full picture of the activities taken inside the system – which actions are performed, who made those changes, etc.

---

## Ransomware protection for cyber resilience and compliance

Since backup is the last line of security against ransomware, it has to be immutable and cyber-proof. To make sure that it is, you need to be attentive to the way the backup vendor processes your data. For example, with GitProtect.io, you can have peace of mind that your data is compressed and encrypted to prevent it from being executed on storage. Thus, even if ransomware manages to access your backed-up data, it won't be able to run and spread on the storage.

For on-premise installations, the agent only receives the authorization data for storage and your Git hosting service while the backup is running. What's more, these details are kept in Secure Password Manager. Thus, in case ransomware infects the device our agent is on, it will have no access to the storage or the authorization information.

However, even if ransomware manages to encrypt your DevOps data, you should have the option to restore your chosen data from any point in time when your data wasn't infected.

What's more, if a backup vendor offers you storage technology that is immutable, and WORM-compliant, so to say, writes each file only once while reading it several times, it will make our DevOps data resistant to ransomware – impossible to modify it or delete.

---

# Disaster Recovery

## Restore and Disaster Recovery – use case & scenarios

When you are facing the need to find the appropriate backup and recovery software for your DevOps environment, you should pay a lot of attention to the Disaster Recovery technology it provides. The main thing is that it should empower you to respond to any possible data loss scenario. While some solutions may offer basic recovery options for your git hosting downtime, true resilience demands readiness for more complex and critical situations inherent in the DevOps landscape.

**Here are the restore options that GitProtect.io provides to help you withstand every disaster scenario possible:**

- point-in-time restore,
- granular restore of the repositories and metadata you urgently need,
- restore to the same or new repository or organization account
- restore to the local device of your choice
- cross-over recovery to another Git hosting platform, for example, from GitHub to GitLab, Azure DevOps or Bitbucket, and conversely)

Moreover, GitProtect.io, unlike other providers, allows you to set up backup plans, monitor their performance, and run a restore of your backups from a single place – a central management console. So, you don't need to install additional apps.

## What if your GitHub/GitLab/Azure DevOps/Atlassian is down?

A Git hosting provider outage is probably one of those circumstances when you need to retrieve your DevOps or Project Management data fast to guarantee your team's uninterrupted workflow. In such a disaster scenario, you should have the possibility to quickly restore your entire DevOps environment from the most recent copy or a specific point in time to your local system (if we speak about GitHub, Azure DevOps, Bitbucket, or GitLab, make sure that you can restore your data as .git).

Among the other options you should have are the possibility to restore your Git hosting local instance – GitHub Enterprise, GitLab Ultimate, Azure DevOps Server, or Bitbucket DC, – or use the cross-over recovery option to restore your software development data to another Git hosting service.

When it comes to Jira with your software development and production data, you should be able to recover your Jira production environment to your local machine, the same or absolutely another free Jira instance.

## What if your infrastructure is down?

Don't forget that the 3-2-1 backup rule is unquestionably one of the most effective backup strategies. What else to say, it has gained widespread acceptance as a data security standard. It's core is that you have at least 3 copies in at least 2 different storage locations with one off-site.

---

As GitProtect.io is a multi-storage system, you can add as many on-premise, cloud, hybrid, or multi-cloud storage instances as you need. What's more, you can easily set up backup replication across them. Thus, you may be certain that you will be able to recover all or only specific data from your second storage device at any time, even if your first backup storage is down.

### **What if GitProtect.io's infrastructure is down?**

As we rely on data security, we must be ready for any event of failure, especially if it threatens our infrastructure. In the case our infrastructure is down, we will provide you with an installer of your on-premise application. After that, all you will need to do is to log in and assign the storage where you keep your copies. That's it, you have access to your backed-up data and can use any of the restore and Disaster Recovery options mentioned above to restore your critical software development and production data.

## **Restore and Disaster Recovery – use case & scenarios**

There are various circumstances when you may need to restore your entire production and/or DevOps environment – service outages, downtime, etc. Restore and Disaster Recovery technologies that you can perform in a click are becoming decision-makers. After all, the main purpose of backup is to enable a fast restoration process for your critical data in the event of a disaster.

What's the easiest way to do it? The ability to restore multiple GitHub, Azure DevOps, GitLab, or Bitbucket repositories and Jira production data.

To make your disaster recovery plan simple, quick, and effective, you should have the possibility to select the repositories or projects you need to restore, view the most current copies, assign them manually, and restore them to your local device, another Git hosting service or, if it comes to Jira production data, to another Jira account.

### **Attention to Jira users**

According to Jira's billing model, you have to pay for each user who uses the application. So, the potential restore of Jira users may seem an issue. Theoretically, restoring your entire production environment may cost twice as much as you do since you will need twice as many people, right? Nope... that's not. With GitProtect.io's no-user recovery option, you can easily restore all your Jira data but for users without exceeding your current Jira pricing plan. Additionally, this feature allows you to restore your data to a free Jira account.

## Point-in-time restore – no limits to the last copy

Human mistakes is a common cause of cybersecurity risks and data loss, they can even lead to outages – in [April 2022 Atlassian Jira experienced a massive outage](#) that lasted for almost 2 weeks. There is no difference if we speak about git repository data or production data – it's always difficult to predict from where the threat can come. You can face unintentional deletion of your critical data, or, in the worst scenario, an intentional one, but still in both cases, all you will need to do is deal with the consequences. Having a backup plan up in your sleeve, you can restore your critical software development and production data with a click, all you will need to know is the time when a disaster took place, and then run your backup copy from the precise moment you need.

It's worth mentioning that most backup providers allow you to restore only the last copy or the copy from up to 30 days prior. But what if you notice some human mistake, let's say, in 6 months? Then, a backup solution with point-in-time restore capabilities and unlimited retention is more than a need.

## Restore to your local machine

Well, GitHub, Bitbucket, or GitLab SaaS versions may be your user option. However, you should have the possibility to restore your copies to your local machine at one time or another. What are the reasons? Service outage, Cloud infrastructure downtime, or poor internet connection. That's why your backup software should allow you to restore your entire Git environment to your local machine in addition to other restore options.

What's more, you never know when the event of failure hits your DevOps environment. So, your backup software should also offer you alternative options – cross-over recovery to another Git hosting provider, restore to the same or new GitHub, GitLab, or Bitbucket repository.

## **No overwriting of repos during the restore process**

Isn't it convenient to have your repository restored as a new one rather than overwriting the original GitHub, GitLab, or Bitbucket repository? You may need to leave the original one for tracking the changes or just keep it for future reference. However, security is the main consideration here. Additionally, in this case, you get full control over your repository, deciding whether you should keep your repo or delete it.

## **Granular restore of only selected data**

There may happen different situations – unintentional deletion, human errors, mistakes in daily operations – when you don't need to restore your entire production or DevOps environment. In this case, you should be able to restore granularly only the data you urgently need. In the case of DevOps, it may be repositories or specific metadata, while in Jira it can be projects, workflows with their dependent elements, and issues attachments.

[GitProtect's Jira Software Granular Restore and Disaster Recovery](#)



Moreover, your backup provider should permit you to recover your software development and production data to the same or a new account, to your local device (or cross-overly to another git service, when it comes to git environments). That's it – you have your data restored in the blink of an eye without interrupting your workflow.

### **Find out more about restore and Disaster Recovery requirements for each of the GitHub, Bitbucket, GitLab, and Jira**

- [Bitbucket restore and Disaster Recovery best practices](#)
- [Jira restore and DR best practices](#)
- [GitLab restore and Disaster Recovery – how to eliminate data loss](#)
- [GitHub restore and DR – scenarios & use cases](#)
- [Azure DevOps restore and Disaster Recovery](#)

---

# Conclusion

Understanding the threats that can impact your production and DevOps environments, whether it's an outage, infrastructure downtime, cyberattack, ransomware, human error, or data breach, it requires a proactive, unified approach from your operations, development, and security teams.

Thus, building a robust and comprehensive data protection strategy for GitHub, Azure DevOps, GitLab, and Atlassian is optional, it's more than critical. **Your security strategy should go beyond basic security testing methods such as SAST, DAST, IAST, or automated security testing.** It should also integrate a broader set of practices and controls to improve the security posture across your organization.

## What are the security components to include?

- Regular DevOps backups and recovery plans
- Software Composition Analysis (SCA)
- Privileged access control
- Secure API management
- Cloud and configuration security
- Secrets management and vulnerability scanning
- Secure code development practices
- Short, frequent development cycles
- Automated software provisioning

## So, what should be your ultimate goal when it comes to security?

Embedding security into every stage of the Software Development Lifecycle (SDLC). It will ensure your organization is equipped to handle modern DevOps threats while maintaining uninterrupted business continuity.

# About GitProtect Lab

GitProtect Lab is a research-driven team of security experts who are eager to uncover and analyze the threats across DevOps and SaaS ecosystems, including GitHub, GitLab, Atlassian, and Microsoft's Azure DevOps. And all of that with the only mission in mind — to help CISOs, DevOps leaders, Jira Admins, and decision-makers strengthen the security posture of their environments, products, and workflows.

At GitProtect Lab, our team of backup experts and threat analysts collaborates to track emerging attack vectors, identify platform-specific risks, and build knowledge in data protection that helps DevOps and security teams worldwide.

Here, you'll find exclusive research, deep-dive reports, and actionable insights — all driven by our mission to make software development and project management safer, more resilient, and compliant.

Want to know more about the future of DevOps security?  
Read GitProtect Lab's latest research on DevOps threats.

---

## About the Author



### Daria Kulikova

Editor / Atlassian Champion

Daria Kulikova is an experienced cybersecurity content strategist and Atlassian Community Champion. With 3+ yrs of background in DevOps security, data protection, and compliance frameworks, she tries to translate complex cybersecurity concepts into actionable insights for tech professionals and decision-makers.

Passionate about secure software development and resilient infrastructure, she actively contributes to security education through reports, dedicated blog posts, and articles.

# About GitProtect Lab

## Meet the Team



### Karolina Rurek-Nowak

Backup Solution Architect with holistic experience in designing and implementing backup and data protection solutions. Supports clients from the public and private sectors at every stage of the process, starting with a deep analysis of security concerns, through technology selection, to solution testing and final implementation.

Specializes in data backup, disaster recovery, and cybersecurity. Works directly with clients and partners, translating business needs to technical requirements and implementing a best-fit cybersecurity solution.



### Jan Rusakiewicz

Product and Project Manager with over 6 years of experience in the enterprise backup and security industry. Jan combines extensive technical expertise with a strong product-focused approach. Since the start of his career, he has emphasized security aspects in software design and development, advocating for secure by design principles and secure coding. In his own words, his goal is to create solutions that significantly improve data security standards and integrate them into DevOps processes across various organizations.



**GitProtect**  
by Xopero ONE


---

## Are you interested in DevOps Backup, Security and Disaster Recovery?

Feel free to reach out to us!



Daria Kulikova

 [/daria-kulikova](#)



Karolina Rurek-Nowak

 [karolina-rurek-nowak](#)



Jan Rusakiewicz

 [/jan-rusakiewicz](#)

[www.gitprotect.io](http://www.gitprotect.io)