

CVE-2022-46169

Cacti Command Injection Vulnerability

Exigent

CISA KEV since Feb 16, 2023

Actively exploited

Public exploit module (Metasploit)

Cacti is an open source platform which provides a robust and extensible operational monitoring and fault management framework for users. In affected versions a command injection vulnerability allows an unauthenticated user to execute arbitrary code on a server running Cacti, if a specific data source was selected for any monitored device. The vulnerability resides in the `remote_agent.php` file. This file can be accessed without authentication.

EVRs

87/100

Immediate priority

CVSS V3.1

9.8

Critical · NVD

EPSS

0.998

99.96th percentile · FIRST

CISA DEADLINE

Mar 9, 2023

KEV since 2023 · listed over a year

Signals behind the score

☒ counted in the score ☐ not counted

evidence for 6 of 8 signals

☒ CISA KEV	Listed in the catalog	☐ Vendor exploitation	Confirmed exploitation in the wild
☐ Ransomware	No known ransomware campaign association	☐ KEV recency	On the CISA KEV catalog for over two years
☒ EPSS	0.998 probability	☐ Pre-patch disclosure	Disclosure status unavailable
☒ CVSS	Base score 9.8	☐ Microsoft Exploitability Index	Not a Microsoft CVE

Remediation

Out-of-band / urgent Patch or fixed version identified**CISA required action:** Apply updates per vendor instructions.

Basis: CISA KEV required action

This function retrieves the IP address of the client via `get_client_addr` and resolves this IP address to the corresponding hostname via `gethostbyaddr`. After this, it is verified that an entry within the `poller` table exists, where the hostname corresponds to the resolved hostname. If such an entry was found, the function returns `true` and the client is authorized. This authorization can be bypassed due to the implementation of the `get_client_addr` function. The function is defined in the file `lib/functions.php` and checks several `$_SERVER` variables to determine the IP address of the client. The variables beginning with `HTTP_` can be arbitrarily set by an attacker. Since there is a default entry in the `poller` table with the hostname of the server running Cacti, an attacker can bypass the authentication e.g. by providing the header `Forwarded-For: <TARGETIP>`. This way the function `get_client_addr` returns the IP address of the server running Cacti. The following call to `gethostbyaddr` will resolve this IP address to the hostname of the server, which will pass the `poller` hostname check because of the default entry. After the authorization of the `remote_agent.php` file is bypassed, an attacker can trigger different actions. One of these actions is called `polldata`. The called function `poll_for_data` retrieves a few request parameters and loads the corresponding `poller_item` entries from the database. If the `action` of a `poller_item` equals `POLLER_ACTION_SCRIPT_PHP`, the function `proc_open` is used to execute a PHP script. The attacker-controlled parameter `$poller_id` is retrieved via the function `get_nfilter_request_var`, which allows arbitrary strings. This variable is later inserted into the string passed to `proc_open`, which leads to a command injection vulnerability. By e.g. providing the `poller_id=id` the `id` command is executed. In order to reach the vulnerable call, the attacker must provide a `host_id` and `local_data_id`, where the `action` of the corresponding `poller_item` is set to `POLLER_ACTION_SCRIPT_PHP`. Both of these ids (`host_id` and `local_data_id`) can easily be bruteforced. The only requirement is that a `poller_item` with an `POLLER_ACTION_SCRIPT_PHP` action exists. This is very likely on a productive instance because this action is added by some predefined templates like `Device - Uptime` or `Device - Polling Time`. This command injection vulnerability allows an unauthenticated user to execute arbitrary commands if a `poller_item` with the `action` type `POLLER_ACTION_SCRIPT_PHP` (`2`) is configured. The authorization bypass should be prevented by not allowing an attacker to make `get_client_addr` (file `lib/functions.php`) return an arbitrary IP address. This

could be done by not honoring the `HTTP\_...` `\$\_SERVER` variables. If these should be kept for compatibility reasons it should at least be prevented to fake the IP address of the server running Cacti. This vulnerability has been addressed in both the 1.2.x and 1.3.x release branches with `1.2.23` being the first release containing the patch.

Affected and fixed versions

PACKAGE	AFFECTED	FIXED IN	SOURCE
cacti	< 1.2.23	no fixed version listed	NVD

Sources

Vendor guidance	github.com/Cacti/cacti/commit/7f0e16312dd5ce20f93744ef8b9c3b0f1ece2216	github.com	github.com/Cacti/cacti/security/advisories/GHSA-6p93-p743-35gf
NVD	nvd.nist.gov/vuln/detail/CVE-2022-46169	CISA KEV	cisa.gov/known-exploited-vulnerabilities-catalog
ENISA EUVD	euvd.enisa.europa.eu/vulnerability/EUVD-2022-48996	Current data	evulnable.com/vulnerability-management/cve/CVE-2022-46169

Snapshot at generation time. EPSS can move quickly. A public exploit module is not evidence of active exploitation.