

# [NT] TOCTOU with NT System Service Hooking

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0003.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 01/01/04

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

Date: 1 Jan 2004 12:40:17 +0200

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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

TOCTOU with NT System Service Hooking

---

## SUMMARY

TOCTOU (Time-Of-Check-to-Time-Of-Use) problem is known for a while [1]. Nevertheless, such bugs are still not uncommon. That is more or less acceptable for general software but not for security products. Andrey believes there are drivers that hook kernel system services by well-known technique [2, 3, and 4]. Those hooks are used to provide mandatory access restriction to registry, files, processes, etc. Of course, all that products suffer from TOCTOU bugs.

## DETAILS

General practice suppose following algorithm

### Technique 1

Hooked service entry point

- 1) Get object name
- 2) Check object name against security policy rules
- 3) If access allowed then call original kernel service function
- 4) If access denied return error code

As object name is in user memory, attacker may change it between steps 2 and 3. So attacker calls system service with object name for which access

## Securiteam: [NT] TOCTOU with NT System Service Hooking

is not restricted. Than object name is simultaneously changed in order to point at restricted object.

Some products seem a bit aware of this problem and use slightly different approach.

### Technique 2

Hooked service entry point

- 1) Call original kernel service function to get object handle
- 2) Get object name from handle
- 3) Check the name against security policy rules
- 4) If access allowed then return success
- 5) If access denied then close the object handle and return error

The approach is also vulnerable, as attacker may use the handle to access object between steps 2 and 4.

Proof of concept:

Andrey has developed small demo driver hookdemo.sys that hooks ZwOpenKey system service in order to prevent any access to 'HKLM\SOFTWARE\hookdemo\test1' registry key. The driver uses both described hooking techniques. After driver started, you have no access on specified registry key by usual means.

Hookcrack tool demonstrates how to successfully bypass hookdemo.sys restrictions.

Both driver and hookcrack as well as their source code may be found at <http://www.securesize.com/Resources/hookdemo.shtml>

To bypass the protection of first kind of hook, hookcrack calls ZwOpenKey with non-restricted key name 'test2'. At the same time separate thread, change the name buffer to 'test1'.

Andrey's tests shown that restricted registry key is accessed in around 5000 iterations, approximately 10 seconds on p4 1.3.

To bypass second hook technique, hookcrack tries to access some registry key to get handle value. That handle value will be the same any time hookcrack starts. The value is saved in generated crack.bat that start hookcrack and supply handle to it as a parameter. Hookcrack calls ZwOpenKey with target key name and at the same time reads that registry key value from separate thread using known handle value.

By this method, Andrey was not able to bypass protection in reasonable time on one CPU machine, but on dual CPU machine the crack succeed just in 3-5 iterations.

Except specified brute force attacks more advanced attacks are possible by using hardware debug breakpoints on memory access.

## Securiteam: [NT] TOCTOU with NT System Service Hooking

Who is affected?

Andrey is not aware particular products suffer this problem but system service hooking is quite popular among developers as it provides abilities that are not present OS's API and proved to be reliable and compatible through different NT versions.

At least, if security product mandatory restricts access to registry on Windows NT prior XP then with high degree of probability it has this bug.

How to fix it?

The problem occurs because checks are used against user mode accessible resources: name buffer and user mode handle. To mitigate this, hooked service must copy user mode OBJECT\_ATTRIBUTES structure with all its sub-buffers to kernel memory. In general, it's not trivial task as requires:

- 1) Composing new name from RootDirectory handle and ObjectName
- 2) Copying security descriptor
- 3) Copying SecurityQualityOfService structure that is undocumented

Once OBJECT\_ATTRIBUTES structure is copied to intermediate kernel memory buffer, the hook must recursively call the same system service. Passing control to original system service merely does not work as 'previous mode' is still set to UserMode and system expects user memory buffers. In case of recursive call, 'previous mode' is set to KernelMode and kernel memory buffers are accepted. It works but opens another worst security hole because system will bypass all security check when call originated from kernel, 'previous mode' set to KernelMode. To close this hole, hook must perform all security checks itself which is quite complicated task to be done correctly and impossible at all in some cases.

Therefore, correct implementation of mandatory security rules by system service hooks is very complicated and rather impossible to achieve. Access to device manager objects (files and devices) may be effectively restricted by standard filter drivers. Unfortunately, it is impossible for rest of objects, such as registry keys, LPC ports, processes, threads. However, XP and w2k3 provide registry callback mechanisms that may be used for access restriction to registry. If you still in doubt how to fix it, look at GeSWall NT framework.

Note, that we stress mandatory restrictions policies implementations. Of course, it is always possible to set standard NT ACLs on any type of objects.

### ADDITIONAL INFORMATION

The information has been provided by <mailto:andr@sandy.ru> Andrey Kolishak.

[1]

<<http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/1996-compsys.pdf>>  
<http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/1996-compsys.pdf>

Securiteam: [NT] TOCTOU with NT System Service Hooking

[2] <<http://www.windowsitlibrary.com/Content/356/06/2.html>>  
<http://www.windowsitlibrary.com/Content/356/06/2.html>

[3] <<http://www.sysinternals.com/ntw2k/source/regmon.shtml>>  
<http://www.sysinternals.com/ntw2k/source/regmon.shtml>

[4] <<http://www.wiretapped.net/~fyre/sst.html>>  
<http://www.wiretapped.net/~fyre/sst.html>

=====

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  
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====  
=====

**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.