

Re: [Full-Disclosure] Microsoft Windows Huge Text Processing Instability

Source: <http://www.derkeiler.com/Mailing-Lists/Full-Disclosure/2004-10/0669.html>

From: James Tucker (*jftucker_at_gmail.com*)

Date: 10/18/04

To: Kaveh Mofidi <admin@securetarget.net>

Date: Mon, 18 Oct 2004 11:57:38 +0100

I am sorry, maybe I just don't get it, but the two forms you are talking about could not happen in the scenario described.

Besides this fact, user data space still has to be violated and this still requires either privileges (which means you have access anyway) or requires an exploit to elevate your privileges (again this makes the vector you describe pointless, it would only cause the ATTACKER issues).

There are some other VERY obvious stupidities here:

1. The system is described as "unstable" and "unresponsive" due to the load and virtual memory usage. Paging takes time and the system seems largely unresponsive whilst waiting for disk I/O. In this scenario, how the * are you to exploit the system? At the point of such high resource load that this becomes a problem the user will be unable to log out. Furthermore the attacker will be unable to log in / run programs unless they have taken control of the PC (not meaning personal computer, look up operating system principles (kernel design, in particular process and memory management) and assembler / machine coding if you dont know what it means).

2. This 'attack' is entirely dependant upon RAM volume. If the system in question has enough physical memory it will remain responsive, as it is paging operations which block (block in the thread sense), not loading operations.

3. `taskkill /f /im "notepad.exe"`

4. Notepad2 and Metapad have been extremely badly coded if it is easier to perform process injection with these than with Notepad. I do not understand why you could not simply take over some other user process or a system process using the same vector. There are plenty of other processes you could use for this purpose.

5. What exactly are you claiming to be "exploiting?"; system load is not an exploit, it's wasteful resource useage. Memory useage quotas are the proper management system for this.

6. Workarounds :-

Replace relevant executables with scripts to prevent loading of files above a certain size.

Educate such stupid people who might try to open a >200mb file in a very basic text editor. What are you planning on doing, reading it all?

Use Wordpad, despite its appearance.

Increase the amount of physical ram available.

Add end task on fail entries into the user control panel section of the registry / via group policy.

this is one of the thins that Telnet is good for. =)

Very well, I agree that you can create a local DoS on low end systems using this abusive process; at the same time you still require access to the system and priviliges. Attacks may only be performed by utilising other exploits or higher level priviliges which already provide the ability to do what you describe (capture data). Using process injection to do this is simply a HUGE waste of your own time.

As for possibilities for remote exploits via this:

- The file will need to be landed first / created upon arrival (slow over a network). Whilst a simple CLI script can create this for you, you need priviliges to do so; in this case you will not be disclosing any information. As described above, the user would not open it themselves more than once (even the most unexperienced people will fear repeating an action which makes most of the system unresponsive (for most users they would consider the load problems as a crash, and will not have the patience to wait for completion).

- If the information in the file contains sensitive data, then why has the user attempted to load it in Notepad? If they created it, it was not in Notepad (as they would have to DoS their machine for several minutes before editing, which would then be painful to use). Somewhat unlikely.

- Remote process injection does not just work because some program is running without a desktop session.

What is the value of this information?

What 'exploit' can really be performed?

Are there not easier ways to achieve the same effect (remember ALL you are doing here is causing mass paging of data, and paging unfortunately blocks in this situation as there is no memory free for the working environment).

Full-Disclosure: Re: [Full-Disclosure] Microsoft Windows Huge Text Processing Instability

On Sun, 17 Oct 2004 11:41:23 +0330, Kaveh Mofidi <admin@securetarget.net> wrote:

- >
- > -----BEGIN PGP SIGNED MESSAGE-----
- > Hash: SHA1
- >
- > Secure Target Network (Security Advisory October 17, 2004)
- > Topic: Microsoft Windows Huge Text Processing Instability
- > Discovery Date: October 14, 2004
- > Link to Original Advisory: <http://securetarget.net/advisory.htm>
- >
- > Affected applications and platforms:
- > Notepad, NotePad2 and MetaPad (Seems like all Text Processing Apps) /
- > Microsoft Windows (All Versions)

You did not test wordpad, but instead tried several thrid party apps instead?
Your assumption that NT is at fault is incorrect. Nothing crashes and
no new vectors are created.

- > Introduction:
- > It is not important, the limitation of opening large text file with
- > "notepad" or similar products like NotePad2
- > (<http://www.flos-freeware.ch>) and MetaPad
- > (<http://liquidninja.com/metapad/>); the point is just the way these
- > tiny text processing apps open and handle large text files (talking
- > about over the 200MB).
- > The way they handle huge text files, it is near possible for a fast
- > modern PC to be completely unstable. This Instability may path to
- > process injection because you cannot even kill the processes of these
- > apps and they will remain "up and running" even when you logged off.

How did you log off? I thought the system was unresponsive? The apps
will close when sent a TERM signal, they just need time to deal with
their memory load.

- > So, it's possible for a unprivileged user to simply hook to the
- > remaining process of a privilege user and this lead to information
- > disclosure (simply reading the content of the memory before swapping
- > a large file which happens time after time, based on the file size)
- > but may even lead to running privileged tasks based on the app they
- > used for processing text.

And how exactly are they going to do that? How much memory useage do
you need for this exploit? To get the system to respond in a timely
fashion you would need to access the PC and slow the paging quantum,
this is by far a trivial task, and contrary to the above this requires
the highest priviliges you can have in in NT, suffice to say not even
Administrators have such access rights.

- > Exploit:
- > It is different to exploit based on the application you choose for
- > text processing; for windows default notepad.exe, it'll be some like

Full-Disclosure: Re: [Full-Disclosure] Microsoft Windows Huge Text Processing Instability

- > *a huge DoS but for NotePad2.exe and MetaPad.exe it is possible to*
- > *doing process injection (information disclosure and/or running*
- > *privileged tasks).*

So what, they open up a shared memory space / revoke their protected user space? No, I suspect they are no easier to exploit than any other; apart from maybe their code is more understandable / better documented, making it easier for the "n00b". Why would you use such a program?

- > *Workaround:*
- > *The best way to work around this situation is just not to open large*
- > *text files in windows! or wait a long time for completion of task.*

No way?

- > *Tested on:*
- > *Microsoft Windows XP SP1/SP2RC2/SP2 on Intel P4 2.4 with 1GB of RAM*

You really installed a copy of each version to test this? :(sux2bu

Full-Disclosure – We believe in it.

Charter: <http://lists.netsys.com/full-disclosure-charter.html>