

**Exploits & Information about Shatter Attacks** 

**Chris Paget <foon@ngssoftware.com>** 



#### **Introduction**

- Software 3 exploits to be released
- Techniques for code injection and execution
- User impersonation and UI manipulation
- Fun stuff to do with GDI messages
- Possible solutions



### **Shatter – The WM\_TIMER Issue**

- Released August 7, 2002
- Accepted as a problem by MS a month later
- Patched in December 2002
- Code injection through an Edit control
- Code execution through WM\_TIMER



# **Smashing – An Exploit**

- Works against any LocalSystem process
- Extensible to other issues
- Bruteforce capabilities
- Code injection through window titles
- Code execution by window handle or thread



### **Smashing Internals**

- Shellcode simply runs Smashing again
- Named pipe communication
- Fails over to a named file
- High-privileged instance creates process
- Bruteforcing is within the limits of SEH
- Code injection through a message box



### **Code Injection – Window Titles**

- WM\_SETTEXT / SetWindowText()
- >500KB of shellcode space
- Shared heap segment (for performance)
- Message boxes from calling process
- Unicode is your friend!
- Window captions are read-only
  - No self-modifying shellcode



# **Code Injection – Alternatives**

- Edit boxes!
- Named pipes
- Files
- Network streams
- Other I/O mechanisms
- Hook libraries



#### **Code Execution**

- Two techniques available
- Direct vulnerability: callback functions
- Indirect vulnerability: UI manipulation
- Indirect techniques dependant upon a buffer overflow or other exception



### **Direct Code Execution**

- Highly reliable
- DefWindowProc jumps to the address
- EM\_SETWORDBREAKPROC
- LVM\_SORTITEMS(EX)
- EM\_STREAMIN / EM\_STREAMOUT
- TVM\_SORTCHILDRENCB
- Hooks



#### **Indirect Execution**

- Cause an exception due to unexpected input
- LB\_ADDSTRING
- LVM\_INSERTITEM
- CB\_ADDSTRING
- Many, many more
- EM\_SETHANDLE screws up the target process, might be exploitable



# **User Impersonation (1)**

- Disable antivirus software
- Example: Disable NAV2003 with:

```
HWND Norton = FindWindow(0,"Norton Antivirus");
HWND Button = FindWindowEx(Norton,0,0,"&Disable");
SendMessage(Norton,WM COMMAND,0x3eb,(LPARAM)Button);
```



# **User Impersonation (2)**

- Anything you like!
- WM\_SYSCOMMAND
- WM\_CHAR
- WM\_LBUTTONUP
- SendInput()



# **UI Manipulation**

- Complete control is possible
- WM\_SIZE
- WM\_MOVE
- WM\_SETTEXT
- CreateWindow()
- WM\_CLOSE
- InsertMenuItem()



# **Consequences for the UI**

- User cannot trust the interface
- Neither can the application
- Any security measure based on the GUI is trivially breakable.
- Example: "Shutdown" button on Logon screen



### **Attack Goals (1)**

- Privilege Escalation
- Many LocalSystem windows on Win2K
- MS02-071 states:

"It's possible for a highly privilege(sic) process to coexist safely with less privileged processes on the interactive desktop", as well as "None of them could be subverted without the WM\_TIMER flaw"

 MS03-025 – privilege escalation in Utility Manager (Windows 2000 only)



### **Attack Goals (2)**

- Interesting trojans / worms
- Scriptable UI redesigner
- Use the UI instead of the API!
- Automatically disable AV / PFW software
- PFW's have another problem IEKill.



#### **Solutions**

- Current Win32 API does not allow for global protection from Shatter attacks
- Architectural changes or API additions will be required
- Applications will probably need recoding
- There's no easy solution
- X11 is good from the ground up



### X11 vs. Win32

- X11 messages are notifications, not commands
- X11 send\_event flag allows apps to filter synthesised events
- Cross-window messages can be disabled in X11
- X11 isn't perfect, but it has advantages



# **Architectural changes**

- Prevent cross-process GDI messages
  - Could use ACLs based on SID or PID
- Permissions-based
  - Admins decide which apps are OK
- Many applications will break (eg PGP)
- Not a complete solution



### **API Extensions**

- Extend GetMessage() to GetMessageEx()
- New structure would identify source
- Source may be PID or X11-style "synthetic" flag
- Applications decide to accept external messages
- Wouldn't break existing software



#### **Problems**

- Architecture change would break lots of software
- API extension would require developers to rework software for new calls
- Existing vuln-by-vuln patching leaves systems open
- Kernel may not actually track message source!



### **Summary**

- Any app with a TreeView, multiline edit box, ListView or RichEdit control is trivially exploitable
- Many other standard controls are vulnerable
- Users can no longer trust their UIs!
- Shatter attacks aren't just for privilege escalation!





**Exploits & Information about Shatter Attacks** 

**Chris Paget <foon@ngssoftware.com>** 

