# 웹성코드

정든품바



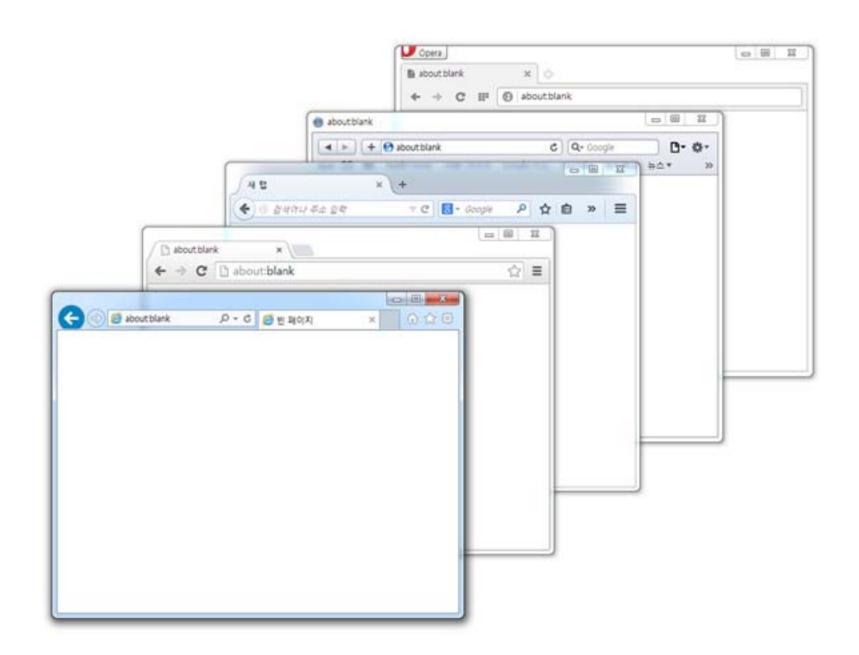


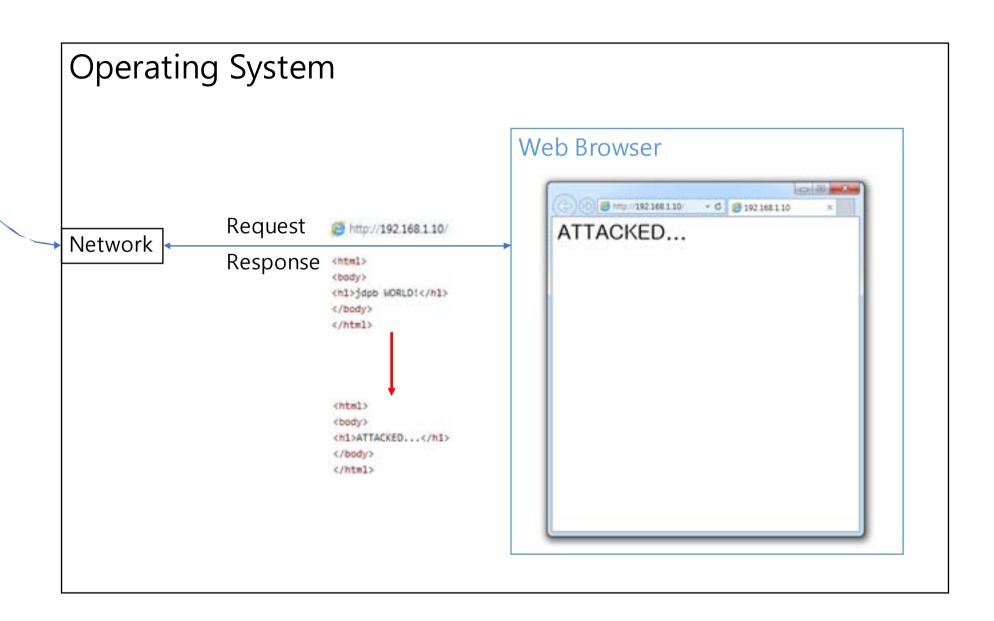
# 웹

보이는 것 그 이상

## 누구나 쉽고 편리하게 그리고 재미있게

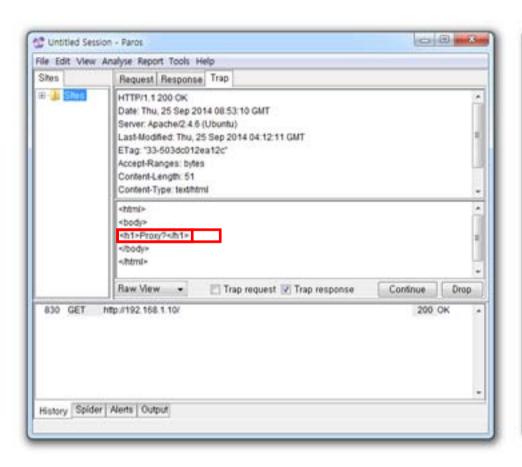
- 생각과 관심만 가지면 누구나 쉽게
- HTML 볼 줄 안다면 누구나 편리하게
- C언어 할 줄 안다면 누구나 재미있게

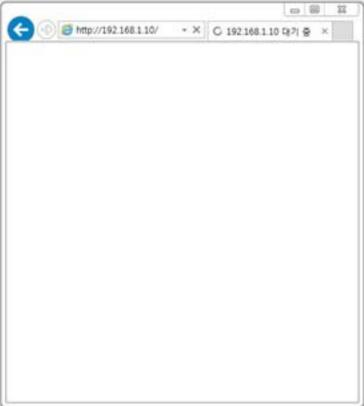




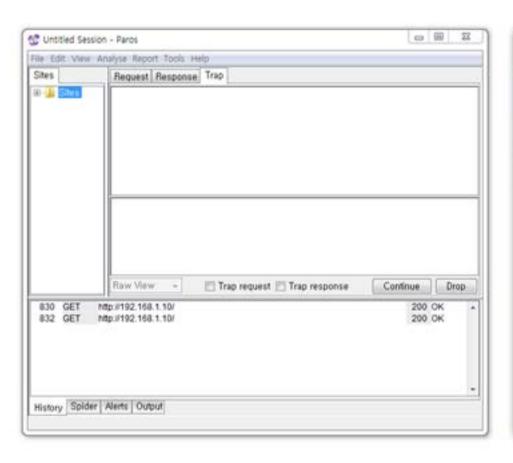
Proxy?

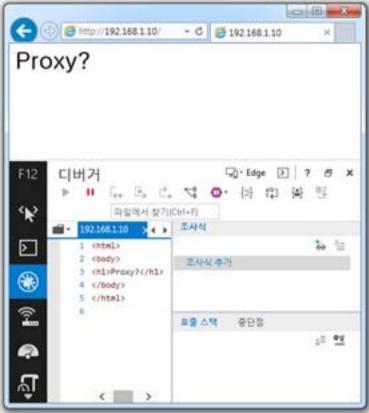
# Proxy



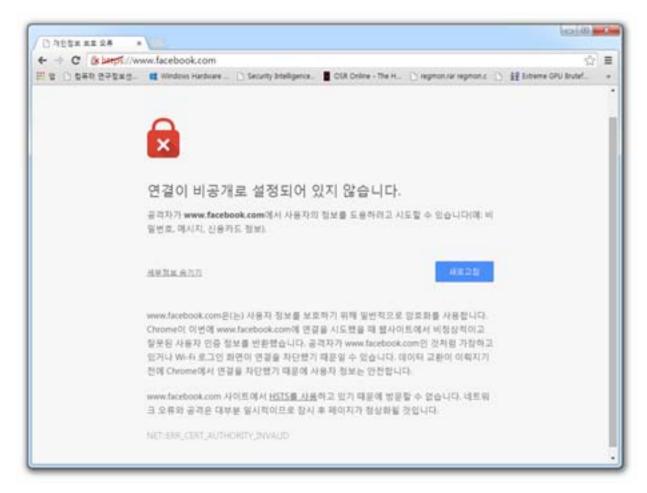


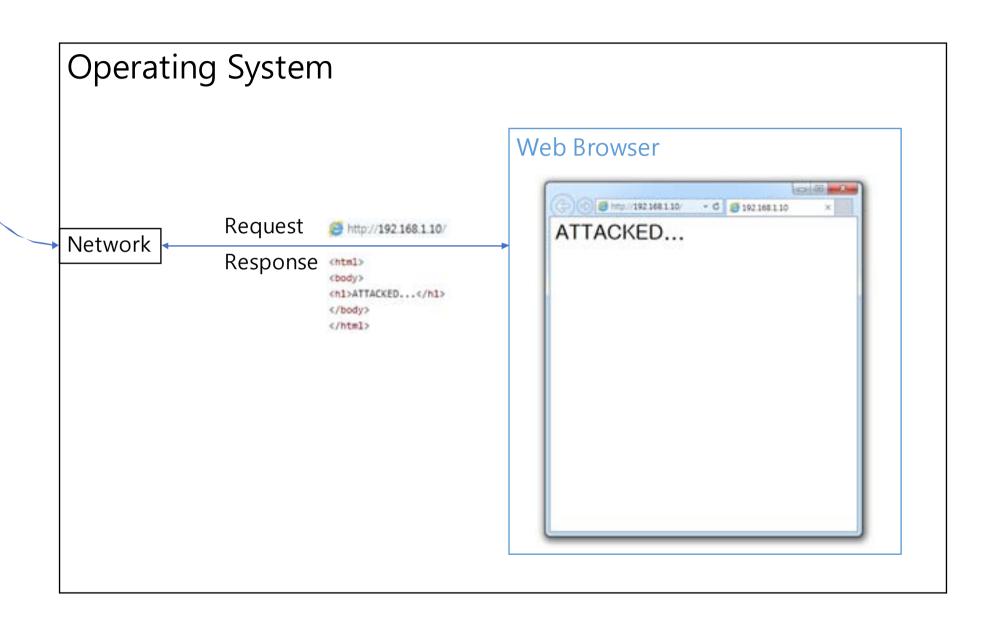
# Proxy





## Proxy (https)





### InternetReadFile function

Reads data from a handle opened by the InternetOpenUrl. PtpOpenFile, or HttpOpenRequest function.

#### Syntax

```
800L InternetReadFile(
_In_ HINTERNET hFile,
_Out_ LPVOID lpBuffer,
_In_ DWORD dwNumberOfBytesToRead,
_Out_ LPDWORD lpdwNumberOfBytesRead
);
```

#### Parameters

hFile [in]

Handle returned from a previous call to InternetOpenUrl. FtpOpenFile. or HttpOpenRequest.

IpBuffer [out]

Pointer to a buffer that receives the data.

dwNumberOfBytesToRead [in]

Number of bytes to be read.

IpdwNumberOfBytesRead [out]

Pointer to a variable that receives the number of bytes read. **InternetReadFile** sets this value to zero before doing any work or error checking.

#### Return value

Returns TRUE if successful, or FALSE otherwise. To get extended error information, call GetLastError. An application can also use InternetGetLastResponseInfo when necessary.

### InternetOpen function

Initializes an application's use of the WinINet functions.

### InternetConnect function

Opens an File Transfer Protocol (FTP) or HTTP session for a given site.

### HttpOpenRequest function

Creates an HTTP request handle.

### HttpSendRequest function

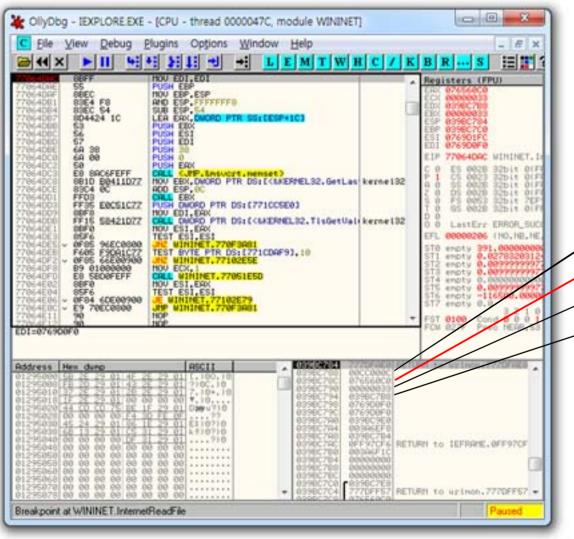
Sends the specified request to the HTTP server, allowing callers to send extra data beyond what is normally passed to HttpSendRequestEx.

# InternetQueryDataAvailable function

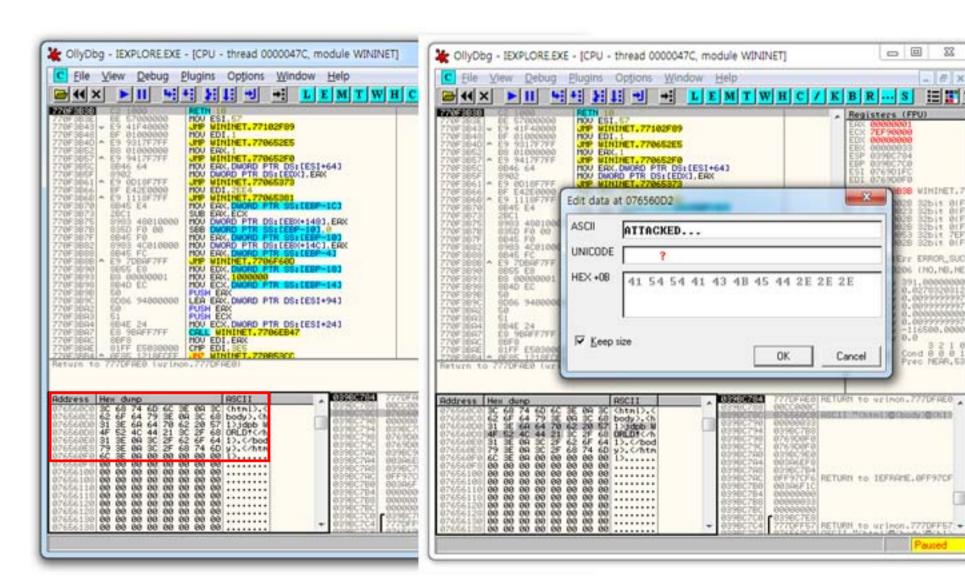
Queries the server to determine the amount of data available.

### InternetReadFile function

Reads data from a handle opened by the InternetOpenUrl, FtpOpenFile, or HttpOpenRequest function.



```
BOOL InternetReadFile(
   _In_ HINTERNET hFile,
   _Out_ LPVOID lpBuffer,
   _In_ DWORD dwNumberOfBytesToRead,
   _Out_ LPDWORD lpdwNumberOfBytesRead
```



0 0

Registers (FPU)

076901F0

EBP

Cancel

RETURN to IEFRATE OFF970F

23

- 6 X

BSB WININET, 7

10bit 01F6 02bit 01F6 02bit 01F6 02bit 01F6 02bit 7FF

32bin 01F

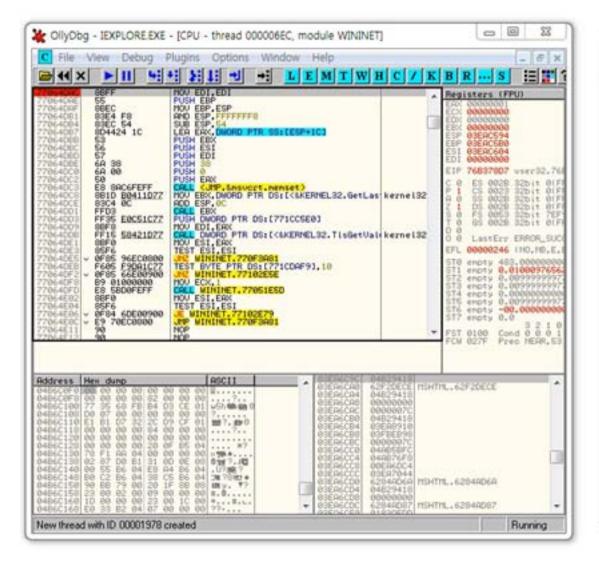
EFF. ERROR\_SUC BOG (NO, NE, NE.

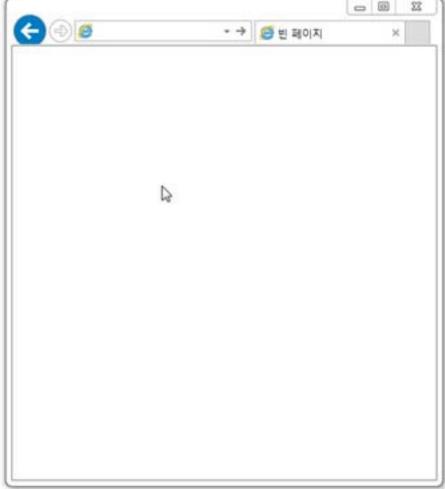
116500,000

Cond B B B

Paused

TWO NEAR, ST





### DebugActiveProcess function

Enables a debugger to attach to an active process and debug it.

#### Syntax

```
BOOL WINAPI DebugActiveProcess(
_In_ DWORD dwProcessId
);
```

### WaitForDebugEvent function

Waits for a debugging event to occur in a process being debugged.

### Syntax

```
BOOL WINAPI WaitForDebugEvent(
_Out_ LPDEBUG_EVENT lpDebugEvent,
_In_ DWORD dwMilliseconds
);
```

#### Parameters

IpDebugEvent [out]

A pointer to a DEBUG\_EVENT structure that receives information about the debugging event.

Value	Meaning
CREATE_PROCESS_DEBUG_EVENT	Reports a create-process debugging event. The value of u.CreateProcessInfo specifies a CREATE_PROCESS_DEBUG_INFO structure.
CREATE_THREAD_DEBUG_EVENT	Reports a create-thread debugging event. The value of u.CreateThread specifies a CREATE_THREAD_DEBUG_INFO structure.
EXCEPTION_DEBUG_EVENT	Reports an exception debugging event. The value of u.Exception specifies an EXCEPTION_DEBUG_INFO structure.
EXIT_PROCESS_DEBUG_EVENT	Reports an exit-process debugging event. The value of u.ExitProcess specifies an EXIT_PROCESS_DEBUG_INFO structure.
EXIT_THREAD_DEBUG_EVENT 4	Reports an exit-thread debugging event. The value of u.ExitThread specifies an EXIT_THREAD_DEBUG_INFO structure.
LOAD_DLL_DEBUG_EVENT	Reports a load-dynamic-link-library (DLL) debugging event. The value of u.LoadDII specifies a LOAD_DLL_DEBUG_INFO structure.
OUTPUT_DEBUG_STRING_EVENT	Reports an output-debugging-string debugging event. The value of u.DebugString specifies an OUTPUT_DEBUG_STRING_INFO structure.
RIP_EVENT	Reports a RIP-debugging event (system debugging error). The value of <b>u.Riptinfo</b> specifies a <b>RIP_INFO</b> structure.
UNLOAD_DLL_DEBUG_EVENT	Reports an unload-DLL debugging event. The value of u.UnloadDII specifies an UNLOAD_DLL_DEBUG_INFO structure.

Meaning

Value

### EXCEPTION\_DEBUG\_INFO structure

Contains exception information that can be used by a debugger.

#### Syntax

#### Members

#### ExceptionRecord

An EXCEPTION\_RECORD structure with information specific to the exception. This includes the exception code, flags, address, a pointer to a related exception, extra parameters, and so on.

### EXCEPTION\_RECORD structure

Describes an exception.

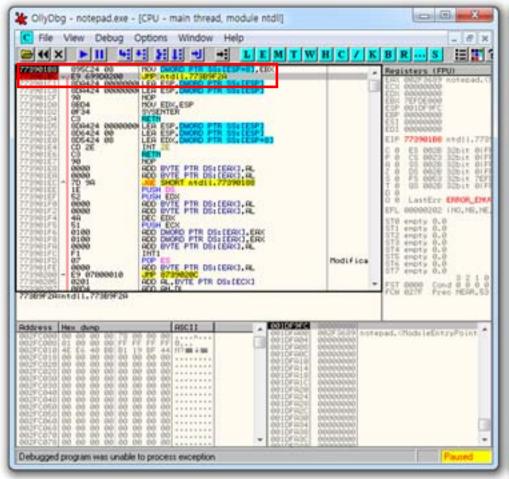
### Syntax

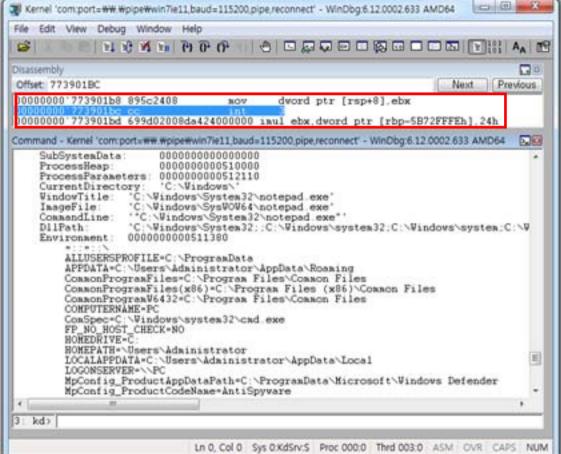
#### Members

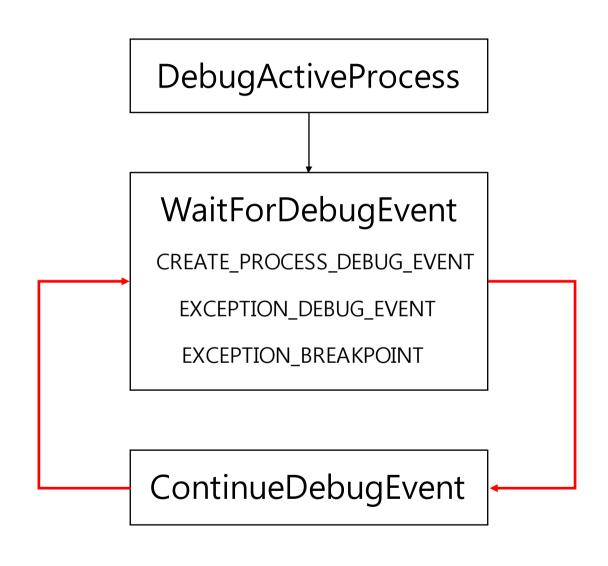
#### ExceptionCode

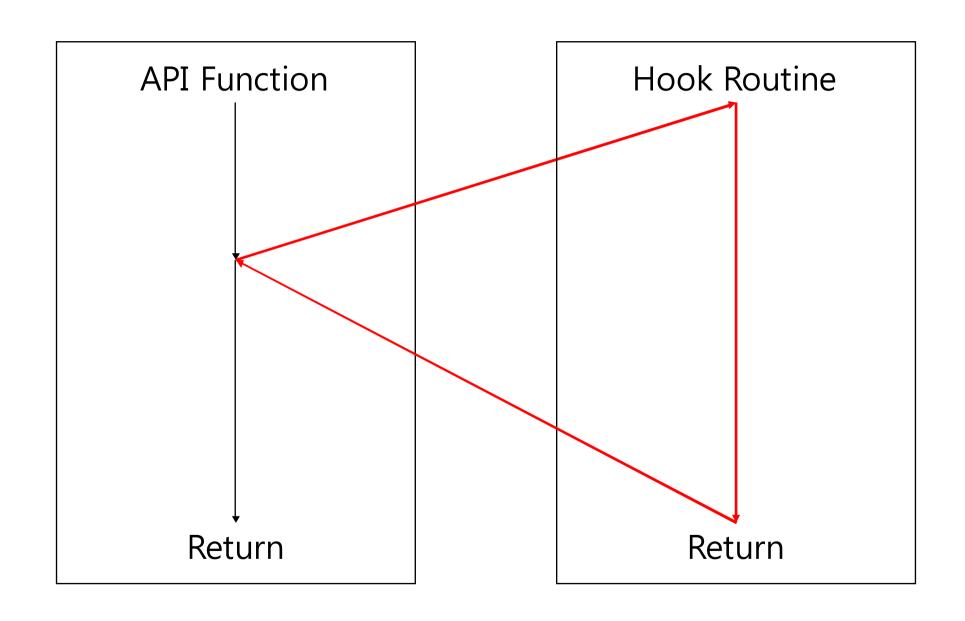
The reason the exception occurred. This is the code generated by a hardware exception, or the code specified in the **RaiseException** function for a software-generated exception. The following tables describes the exception codes that are likely to occur due to common programming errors.

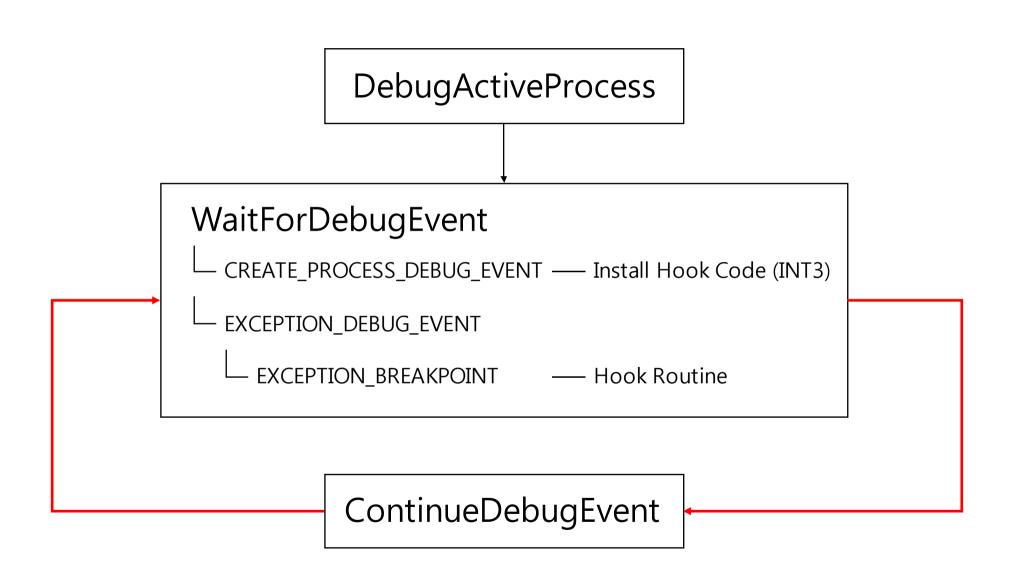
Value	Meaning
EXCEPTION_ACCESS_VIOLATION	The thread tried to read from or write to a virtual address for which it does not have the appropriate access.
EXCEPTION_ARRAY_BOUNDS_EXCEEDED	The thread tried to access an array element that is out of bounds and the underlying hardware supports bounds checking.
EXCEPTION_BREAKPOINT	A breakpoint was encountered.
EXCEPTION_DATATYPE_MISALIGNMENT	The thread tried to read or write data that is misaligned on hardware that does not provide alignment. For example, 16-bit values must be aligned on 2-byte boundaries; 32-bit values on 4-byte boundaries, and so on.
EXCEPTION_FLT_DENORMAL_OPERAND	One of the operands in a floating-point operation is denormal. A denormal value is one that is too small to represent as a standard floating-point value.
EXCEPTION_FLT_DIVIDE_BY_ZERO	The thread tried to divide a floating-point value by a floating-point divisor of zero.
EXCEPTION_FLT_INEXACT_RESULT	The result of a floating-point operation cannot be represented exactly as a decimal fraction.
EXCEPTION_FLT_INVALID_OPERATION	This exception represents any floating-point exception not included in this list.
EXCEPTION_FLT_OVERFLOW	The exponent of a floating-point operation is greater than the magnitude allowed by the corresponding type.
EXCEPTION FIT STACK CHECK	The stack overflowed or underflowed as the result

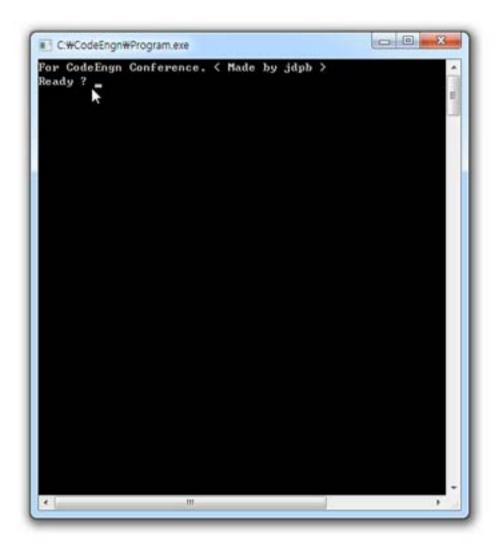














```
<html>
      <form ~
            <input type="text ~
            <input type="password"~</pre>
ID:
                  PW:
                                     Login
```



## Web Browser

```
<html> <script> <br/> ... </html> </script>
```

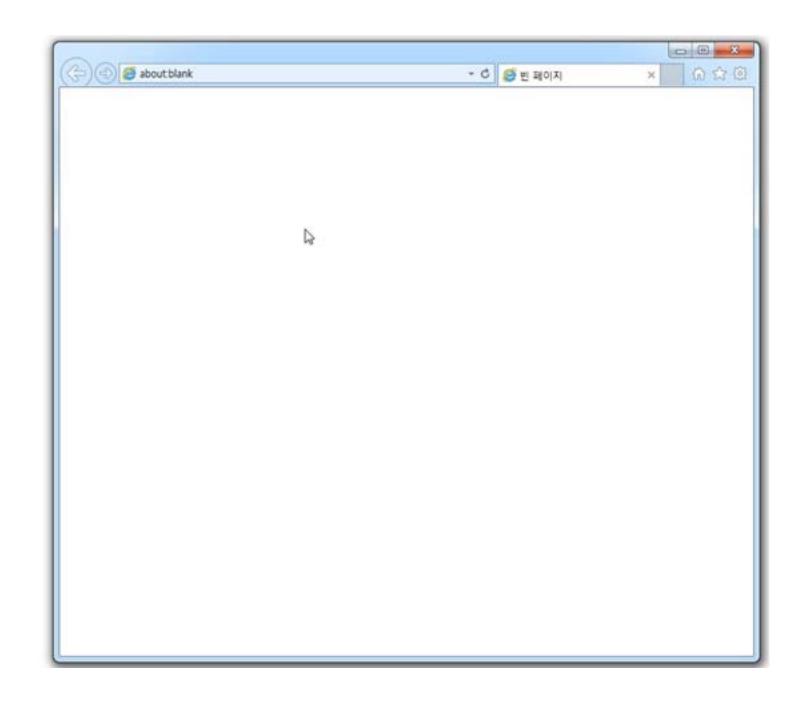
## Web Browser

## Web Browser

Page Read

Script Read

## **Operating System** jdpb.js Web Browser Hook function CodeEngn() alert('jdpb WORLD!'); <html> <script ~ src="jdpb.js"> jdpb.js function CodeEngn() alert('ATTACKED...');

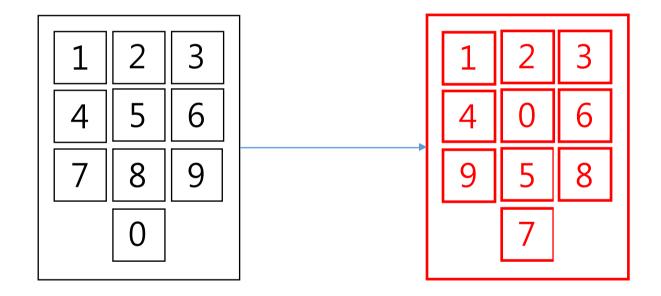




```
<html>
     <map ~
          <area onmousedown= ~
                               KEYPAD HASH VALUE ++
```

### <area onmousedown= ~

. . .



# Thank You

