Chapter 12 - Client Side Attacks
How does this work?
http://www.bobstechtalk.com/blog/2010/4/8/example-exploiting-a-typical-windows-domain-network.html

How did we get around the firewall?
-Reverse shell on HTTPS 443.
-netstat -an | find "4444"
-python Simple HTTP Server

How do we know what the client is to exploit?
-HTTP Headers
-https://en.wikipedia.org/wiki/List_of_HTTP_header_fields

How do we control the client?
-If you want to simulate someone clicking on a link, use location.href
0If you want to simulate an HTTP redirect, use location.replace

Heap Spray
- Stack v. Heap

Stack is used for static memory allocation and Heap for dynamic memory allocation, both stored in the computer's RAM .

Variables allocated on the stack are stored directly to the memory and access to this memory is very fast, and it's allocation is dealt with when the program is compiled. When a function or a method calls another function which in turns calls another function etc., the execution of all those functions remains suspended until the very last function returns its value. The stack is always reserved in a LIFO order, the most recently reserved block is always the next block to be freed. This makes it really simple to keep track of the stack, freeing a block from the stack is nothing more than adjusting one pointer.

Variables allocated on the heap have their memory allocated at run time and accessing this memory is a bit slower, but the heap size is only limited by the size of virtual memory . Element of the heap have no dependencies with each other and can always be accessed randomly at any time. You can allocate a block at any time and free it at any time. This makes it much more complex to keep track of which parts of the heap are allocated or free at any given time.

- JavaScript Heap Spray

Heap sprays for web browsers are commonly implemented in JavaScript and spray the heap by creating large strings. The most common technique used is to start with a string of one character and concatenate it with itself over and over. This way, the length of the string can grow exponentially up to the maximum length allowed by the scripting engine. Depending on how the browser implements strings, either ASCII or Unicode characters can be used in the string. The heap spraying code makes copies of the long string with shellcode and stores these in an array, up to the point where enough memory has been sprayed to ensure the exploit works.

ASLR - Address Space Layout Randomization

Address space layout randomization (ASLR) is a memory-protection process for operating systems (OSes) that guards against buffer-overflow attacks by randomizing the location where system executables are loaded into memory.

DEP - Data Execution Prevention

DEP enables the operating system to mark one or more pages of memory as non-executable. Marking memory regions as non-executable means that code cannot be run from that region of memory, which makes it harder for the exploitation of buffer overruns. DEP prevents code from being run from data pages such as the default heap, stacks, and memory pools. If an application attempts to run code from a data page that is protected, a memory access violation exception occurs, and if the exception is not handled, the calling process is terminated.

12.3.1 - Question 2:

Some of the tactics I would employ (as I believe the question is assessing) are the ones which target the human vulnerability as this exploit fundamentally does. One tactic would be to introduce a "legitimate" page in the background, as to have the client focus on that as opposed to a common "pop up" that bores them. Another I think would help would to be change the certificate settings and applet name to "Google Inc." or some reference to the site you are accessing. Alternatively, making it look less believable (which in my opinion is a valid exercise when determining the awareness of users during a penetration test) would be having no reference to a page, and with weird alarming names such as "Hacked Applet" etc. Wondering if there was anything else anyone could attest to? I hope this is along the lines of the requirements?