EECE 412, Fall 2008

Quiz #4

This quiz consists of 6 pages. Please check that you have a complete copy. You may use both sides of each sheet if needed.

Your Family name: _____

Your Given name: _____

Your student ID: _____

Name of your left neighbor: _____

Name of your right neighbor: _____

#	Points	Out of
1		4
2		6
3		3
4		9
5 (bonus)		3
TOTAL		22

ATTENTION: When necessary, make reasonable assumptions and state them clearly in your solutions.

1. Consider the following example code in C.

```
void foo (int a, char* s) {
        char buffer[10];
        strcpy(buffer, s);
}

void main( int argc, char* argv[ ] ) {
        foo(1, argv[1]);
}
```

If everything goes fine when function foo is called, then the memory layout during execution of foo is shown in the following figure, where the thick black arrow shows how the program counter would change on the return from foo to main.

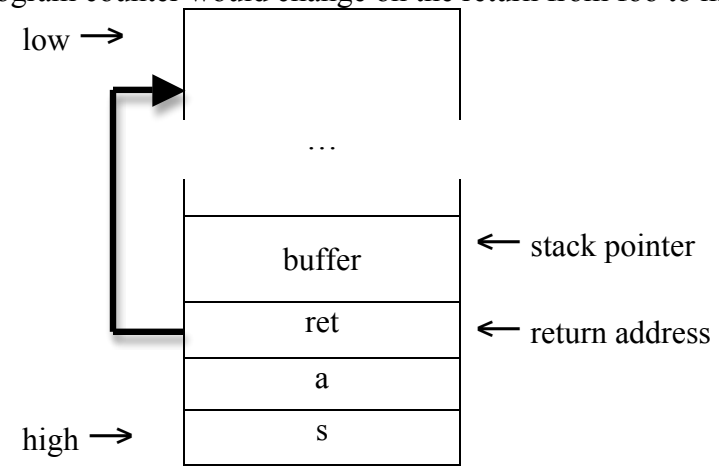


Figure 1.

Now, suppose a buffer overflow has occurred in foo, which resulted in the following memory layout. Such an overflow would generally crash the program.

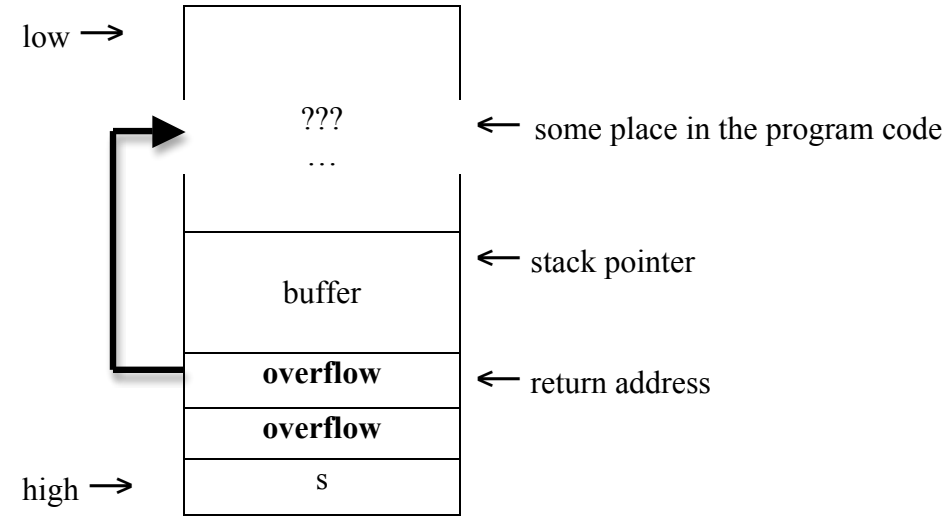


Figure 2.

Now, imaging that at another time, when the program was executed again, another buffer overflow (this time more malicious) occurred, which resulted in the memory layout shown in the following Figure 3.

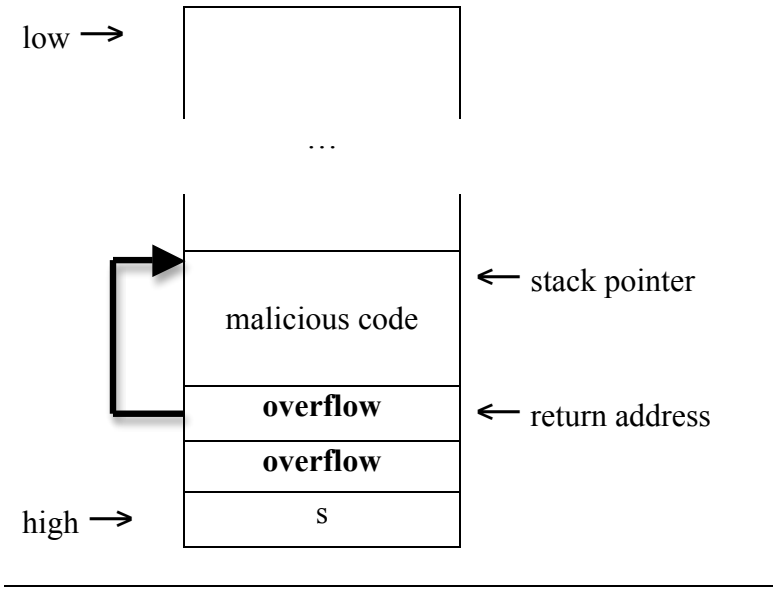


Figure 3.

Suppose "no execution" (NX) bit method of protecting against buffer overflow is implemented at the OS and the underlying hardware where the above program runs in both of the following questions.

a. Could the buffer overflow illustrated in Figure **2** succeed? Explain why.

Yes, because NX bit method would not prevent from return address modification.

b. Could the buffer overflow illustrated in Figure **3** succeed? Explain why.

No, because NX bit method would prevent the malicious code from being executed.

2. Provide three examples, one for each, of the following techniques employed for creating malware signatures and fingerprints. **Explain why each example illustrates the corresponding technique.**

a. Example of content analysis

Analysis of errors propagated from one malware instance to another. When fragments/parts of malware are copied, errors tend to be copied as well. This enables to "fingerprint" malware and track it back to the original source.

Another example is the style of naming functions and other malware modules. Same writer(s) of malware tend to use similar/same names or at least naming styles from one project to another.

b. Example of non-content analysis

Length of functions code in the malware is a non-content trait that is followed by malware writers from project to project.

c. Example of individual or group identification

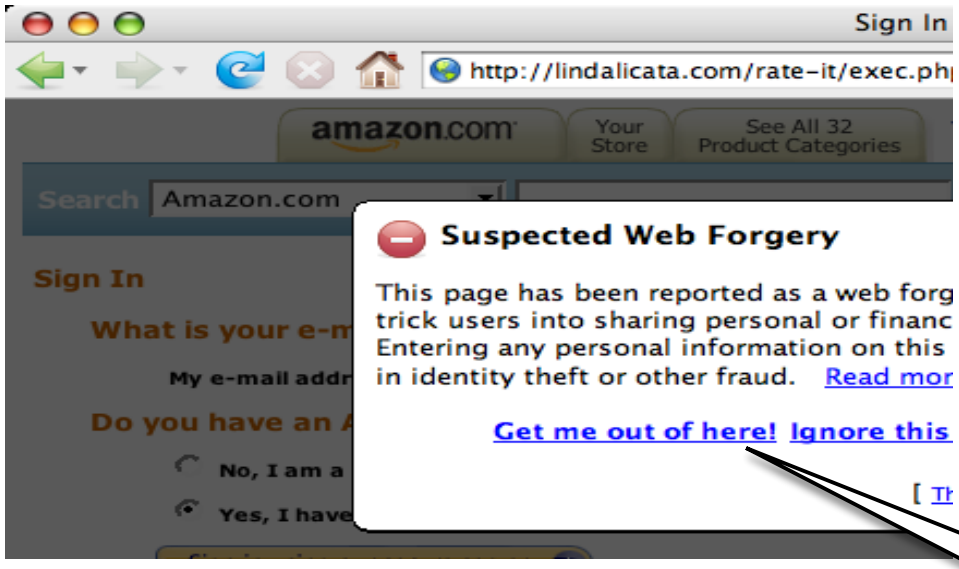
Group or individual malware writers tend to develop malware that runs on specific hardware or OS platforms just because the writer(s) happened to have access to exactly this platform(s). Another example is the use of specific GUI or other convention(s) that tend to be used consistently from project to project.

3. Explain the difference between **verbose** and **blind** SQL injection attacks.

In the case of verbose (a.k.a. normal) SQL injection, the attacker relies on SQL error messages to piece together the SQL code being executed, enumerate backend database tables, columns, etc. If the application does not return error messages, then the attacker has to resort to blind SQL injection, when the application's behavior is used to identify SQL injection possibilities. Specifically, the attacker asks the server a series of true/false questions and builds up her results from the answers.

4. In the class on Usable Security, we discussed the study by Cranor et al. on the effectiveness of three browser phishing warning mechanisms (IE passive, IE active, and Firefox). Firefox was found to be more effective than either of IE's warning schemes. Google's new Chrome browser has also implemented a phishing warning. Below are the phishing warnings for Firefox (top) and Chrome (bottom).

FireFox:



Google's Chrome:

In case it's hard to read from the above screenshot, the text of the warning message in Chrome is as follows: Warning: Suspected phishing site! The web site at www.casos.diagnosticoveterinario.com has been reported as a "phishing" site. Phishing sites trick users into disclosing personal or financial information, often by pretending to represent trusted institutions, such as banks. Learn more about phishing scams.

Make comparisons of Chrome with Firefox for the following aspects of usability:

a. Capturing the attention of the user.

Chrome should be the same as Firefox – both interrupt the user and prevent him from completing his primary task until dealing with the warning.

b. Helping the user avoid dangerous errors.

Firefox is better – the default action (the left link) is to NOT go to the website, while the default button in Chrome (left button) would take the user to the phishing web site.

c. Educating the user to recognize phishing sites

The Chrome browser emphasizes the domain name in bold type – this should help users begin to recognize that the site is not the site they meant to go to (one key aspect we learned in CMU's phishing game).

5. **Bonus question**: If the study mentioned in the previous problem was replicated, do you think Chrome would perform better than Firefox? Explain your answer.

No, it would not. None of the study participants that saw the Firefox warning was tricked into giving information to the bogus site (45% of IE active did) – Chrome can't perform better than 0% ;-)