

Name: _____ Student ID: _____

CSE 227 Computer Security
Winter Quarter, 2008
Final Exam

Instructor: Stefan Savage

Name _____
Student ID _____

Attention: This exam has 5 sections and you must answer all of them. You are expected to finish the exam in no more than three hours and then send me a completed exam (either using this pdf file, or as a separate text file via e-mail). You are free to use your papers, my presentations, your notes, any books you have around and limited Internet use (limited to the links I provide here and material on the class Web page). I'm trusting you to abide by this, and I think it should be easily feasible.

As with any exam, you should read through the questions first and start with those that you are most comfortable with. Be sure to answer **all** parts of each question you answer. If you believe that you cannot answer a question without making some assumptions (common in my questions), state those assumptions in your answer. .

1	
2	
3	
4	
5	
Total	

Name: _____ Student ID: _____

1. Anonymous communication

In class we discussed the challenges posed in realizing anonymous communication, however we focused on a particularly narrow – albeit popular – part of the design space: overt interactive anonymous packet delivery (e.g., Web browsing via ToR). Under this model, a local eavesdropper can determine that you are interested in communicating privately, but not to whom, non-local eavesdropped cannot identify the sender and individual packets are routed through the network quickly enough to support interactive applications. For this question consider a different part of the design space: covert non-interactive anonymous document delivery. Here you do not care about interactive response time, so its ok to be relatively slow, but you also do not want it to be known that you are sending or receiving documents anonymously.

- a) Sketch out how you might design a system with such properties by piggybacking on the existing USENET news system (USENET, also called netnews, distributes billions of articles a day by flooding messages across all participating servers).

Note: if you want to understand how it works in somewhat more detail you can visit:

http://www.tcpipguide.com/free/t_TCIPNetworkNewsTransferProtocolNNTP.htm or

<http://www.netwinsite.com/dnews/guide.htm>. For absurd detail: <http://tools.ietf.org/html/rfc3977>

- b) Describe how easily an adversary could violate the design goals of this system and compare this to the capabilities required to compromise ToR.

Name: _____ Student ID: _____

2. Geometry Attacks

Hovav's paper, "*The Geometry of Innocent Flesh on the Bone*" describes a technique for constructing malicious software by diverting control flow into various snippets of code (frequently jumping into the middle of existing instructions) that do some small work and return. From these "gadgets" he shows that this "return-oriented" computing model is completely general and that actual code injection is unnecessary for an adversary to implement arbitrary programs. Dean Tullsen is asked by Intel to design architectural changes to the x86 processor to defend against such attacks and he comes up with three proposals. For each describe what you think it is trying to do to help the problem, under what circumstances it will or won't work, and how the adversary might get around it (if possible)?

- a) Procedure Cookies. Like StackCookies, here the idea is that when an executable is loaded into memory, the first 64bits of each of its function entry points is XORed with a random 64bit magic cookie (selected when the process starts). In turn, when the processor executes a `call` instruction it transparently XORs the first 64bits of the target with this cookie to produce the correct instruction to execute.

- b) Function Zones. As an executable is loaded into memory, the symbol table is used to generate an in-memory address map indicating where each function starts and ends in memory. The processor uses this map when fetching instructions to execute and will only fetch instructions that either i) are in the body of the currently executing function or ii) are the *first* instruction of some other function.

- c) Short Return Sequence Restriction. The processor generates an exception if a return is executed within 6 instructions of any unconditional control flow (the standard Windows function epilog is > 6 instructions).

Name:_____ Student ID:_____

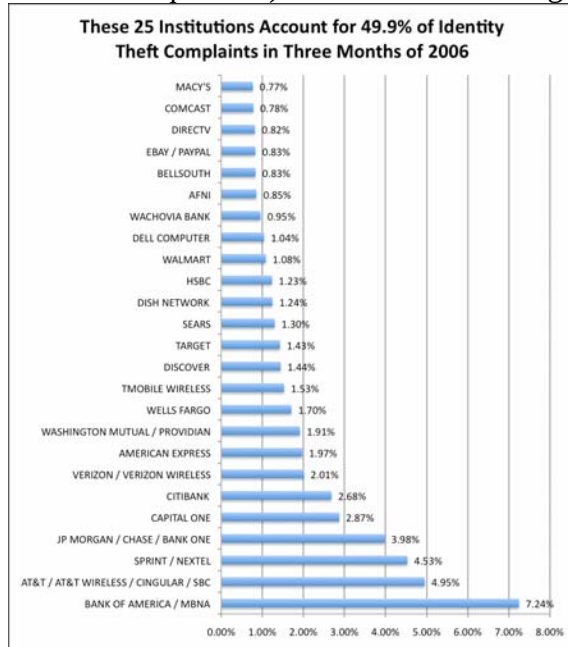
3. Polymorphic Spam

In class we've talked about the arms race between attacker and defender that pervades much of software security. For example, the Polygraph system described techniques for identifying signatures for worms/viruses that were polymorphic in nature. Another domain in which such an arms race occurs is in Spam, or unsolicited bulk e-mail. Consider whether the Polygraph approach could be applied to filtering spam instead of malware. Describe how the spam filtering is similar and different from malware signature generation, how Polygraph's algorithms would apply or need to be changed, and how you might expect spam to evolve if such a defense were developed and deployed.

Name: _____ Student ID: _____

4. Measuring Fraud

In a recent paper, Chris Hoofnagle, used citizen complaint data obtained from the Federal Trade Commission to measure the incidence of identity theft among various banks and institutions (you can find the paper here: <http://repositories.cdlib.org/bclt/lts/44/> although I do not believe you need it to answer this question). Here is one of the graphs from his paper:



Consider the strengths and weaknesses of this measurement methodology when compared to the measurement of compromised bank/credit accounts discussed in Franklin et al. Are they both measuring the same phenomenon? Where do the sources of bias in each approach come from?

Name:_____ Student ID:_____

5. Exfiltration Prevention

The new hot market in the network security appliance market is no longer intrusion prevention (detecting and stopping being from breaking into the network) but exfiltration prevention (preventing intellectual property or sensitive data from leaving the network).

- a) Most of these systems are of the “permit by default” variety. The administrator identifies restricted data that should not leave the company (e.g. social security numbers, secret documents, etc) and then the network appliance scans all outbound network flows (including file transfers, e-mail attachments, etc). Encrypted communications are only allowed if the key has been escrowed with the appliance (so it can read the clear-text version of the content) and flows which cannot be parsed or have very high entropy are blocked unless the administrator give permission. Based on what you know from the class, describe the ways in which an adversary might bypass such systems and how hard it would be to exfiltrate large amounts of data.
- b) Another approach, exemplified by the Glavlit system built at UCSD, operate under the “deny by default” regime. Here the network administrator identifies the particular content that is allowed to leave the network and scans all outgoing connections (in the same manner as above) to ensure that network flows only comprise this content. All other network traffic is blocked. Based on what you know from the class, describe the ways in which an adversary might bypass such systems and how hard you think it would be to exfiltrate large amounts of data.