CS 491/591 Spring '09 Test 4

Name:_____Key_____

Note: There are no correct or incorrect answers on test 4, but this is how the instructor would have answered the five questions in a non-perfunctory way.

#1. In lab 2, we saw an example of how "information only has meaning in that it is subject to interpretation." A magic number (58623) was first interpreted as a signed 16-bit integer (-6911) to check the array bounds, then as an unsigned integer (58623) as the amount of data to read in a buffer. Then after the buffer was overflowed, this 16-bit data value was interpreted as the machine code for JMP ESP (0xff 0xe4) to jump to the stack pointer. In fact, in your lab2exploit.c code the data started out as an ASCII string (0x35 0x38 0x36 0x32 0x33) which was interpreted as input into an unnamed pipe and became part of a TCP/IP socket payload, *etc.*, so the same data was interpreted in dozens of ways during your exploit.

Name another example of information "only having meaning in that it is subject to interpretation" that came from another lab besides lab 2. It has to be another lab, something we talked about in lecture will not be accepted as an answer.

In lab 3, the raw packets sent and received during nmap's probes are subject to interpretation by many different actors, including firewalls, routers, the victims of the port scan, and zombies used for idle scanning. Consider an idle scan. What defines what an attacker can or cannot do with this type of scan? They usually cannot see the responses to probes if they don't put their own return address, because routers on the return path base their routing decisions on the bit pattern that the attacker had put in the source IP address field. They can, however, tell if a zombie machine received a response to one of their probes by interpreting the IPID fields of subsequent packets from the zombie as evidence of whether or not the zombie has been sending packets to other machines (which with some probability could be the victim).

#2. Von Clausewitz said, "Where two ideas form a true logical antithesis, each complementary to the other, then fundamentally each is implied in the other. If the limitations of our mind do not allow us to comprehend both simultaneously, and discover by antithesis the whole of one in the whole of the other, each will nevertheless shed enough light on the other to clarify many of its details. In consequence we believe that the earlier chapters about defense will have sufficiently illuminated the aspects of attack on which they touch. But this is not always so. No analytical system can ever be explored exhausti vely. It is natural that, where the antithesis does not lie so close to the root of the concept as in the previous chapters, what we can say about attack will not follow directly from what was said there about defense. A shift in our viewpoint will bring us nearer the subject, so that we can examine more closely what we previously surveyed from a distance. This will supplement our previous analysis; and what will now be said about attack will frequently also cast more light on defense."

(This is the intro to the part of the book that is about attack, which comes right after the part that is about defense.)

In the context of computer and network security, name something about defense that you have learned more details about by carrying out the attacks in one of the labs. Your answer should relate directly to one of the labs.

By hashing out all of the details necessary to complete the second part of lab 4, which required the forging of raw TCP/IP packets, students should have learned a lot of the layer 3 and layer 4 details that make layer 7 filtering harder than most people realize. It's easy to read a tutorial about Snort or another IDS and see that IDSes have to worry about packet reassembly and packet reordering to be effective at matching signatures in layer 7. However, the problem goes much deeper than this and is a fundamental problem that applies to a broader context. There is a semantic gap between how packets get interpreted by an IDS or other network device and how packets are interpreted by various end hosts. This means that <u>any</u> attempt to limit the communications between end-hosts by some device that only sees the TCP/IP packets passing between them has a necessarily incomplete and possibly skewed view of what is happening. This is true of worm and attack filtering, censorship, VoIP monitoring, intellectual property leak detection, provenance systems, p2p traffic engineering, and many other things.

#3. Randy Browne, in his paper on the Turing Test for Information Flow, defined entropy to be "mathematical uncertainty about the definition of a system," rather than true uncertainty. In your opinion, would it be possible to build a system that enforces the Bell-LaPadula model of confidentiality and stops all forms of information leaks that violate policy, including <u>all</u> covert channels. Your answer should relate to Randy Browne's definition of entropy in some meaningful way.

It might be possible to do so, including stopping all covert channels. The concept of entropy that is stated in the question is very insightful in that we can always miss a covert channel because something we consider to be entropy for the attacker in the form of noise (electrical noise and air turbulence as a hard drive spins, the electrical and heating properties of the chips and motherboard, etc.) could be modeled by the attacker so that they recognize a cause and effect chain we missed that they can use as a covert channel.

However, this definition of entropy does not apply to all scales of the physical world. Feynman proposed a "reversible computer" that never destroys information and could theoretically compute with no energy if given an infinite amount of time. No energy leaking would mean no information leaking, so possibly a system could be built at the quantum scale that had theoretical bounds on its information leakage and computed results in finite time (by applying Moskowitz and Kang's short message criterion, which states that covert channels are okay as long as you can bound the total amount of information they can leak instead of just the rate, so that more time doesn't buy the attacker anything).

At the speed of light, the attacker could perhaps be shown a different computation result than the receiver of the real output using the theory of relativity. Maybe, I don't really know the physics well enough to know if this would work.

In short, the security of the system could be grounded in physics and while we couldn't stop all information flow, we could put hard bounds on it using the short message criterion.

#4. There are two important concepts (actually, there are many, but these are two of the most important ones) that we can learn from Claude Shannon: (1) The purpose of communication is to affect the behavior of the receiver, and (2) the information capacity of a channel is quantitatively equivalent to the receiver's uncertainty of the signal that will be received.

What do each of these concepts have to do with nmap and lab 3?

Lab 3 should have given students some idea of how keeping the configuration of networks, firewalls, etc. secret from an attacker is a deeper problem that goes well beyond the current capabilities of nmap. What port scanning capabilities will the attackers of tomorrow have? Can an attacker do indirect (e.g., like an idle scan) OS detection? Can they learn things through other channels of uncertainty such as limited buffers, variations in packet processing time, or inference channels in the victim's kernel implementation? We saw that an attacker can send a probe to affect the behavior of the victim and then learn information based on their uncertainty of how the victim will respond to the probe. But if we apply these general ideas in a much broader sense, we can think about much more advanced attacks than are available in the current version of nmap.

#5. What do Internet censorship and Metalica's music have to do with each other (other than that both of these things keep getting worse)? You can relate some concept in music to TCP/IP filtering and talk about interactions between multiple layers of abstraction. Or you can discuss Metalica's advocacy against pirated music and Napster, and how Digital Rights Management is or is not a form of censorship. This question is open ended but your answer needs to be serious, silly answers will not be accepted. Bonus point if you directly tie your answer into a specific song from Metallica's good (*i.e.*, pre-1990) albums.

I'm not going to give an answer for this one since it's a slightly political question. Instead, I'll just give you an interesting quote and then the best student answers I saw on both sides of the argument of whether DRM=censorship (the question was open-ended enough you didn't have to take a side on this, but most students argued one way or the other).

"On the one hand information wants to be expensive, because it's so valuable. The right information in the right place just changes your life. On the other hand, information wants to be free, because the cost of getting it out is getting lower and lower all the time. So you have these two fighting against each other."

-Stewart Brand, 1984

The best argument for DRM=censorship that I saw was that the government can't enforce digital rights without giving themselves the power to view all communications between people, which is contrary to the freedom of speech.

The best argument against DRM=censorship that I saw was that censorship is about the <u>availability</u> of information. Requiring payment for content and trying to enforce that is vastly different than making the content totally unavailable.