CSE 551

Design Exercise #1
A Virtual Machine Monitor for the Internet

First draft due: noon, Thursday, April 9, 2009
Final draft: 4:30pm, Thursday, April 16, 2009

An operating system, such as UNIX, provides several key pieces of functionality for its users: an abstract virtual machine execution environment for running applications, strong isolation between different users and different applications, persistent storage with access control, and so forth. The Internet protocol suite is akin to an operating system for the network, but many traditional OS services are lacking from the Internet, even today. Specifically, the designers of the Internet intentionally ignored security, and thus they put very little effort into mechanisms to isolate users from each other.

An extreme form of an operating system (which we'll read about later in the quarter) is called a virtual machine monitor (VMM). Instead of providing applications an abstract virtual execution environment, a VMM provides an emulation of the physical hardware. In this way, a VMM can run an operating system as an application, and if the OS it runs is itself a VMM, that OS can run an OS as an application. (A VMM is usually implemented using virtual memory tricks, but that is less important for our purposes.) The benefits of such an approach are several. First, it provides a way of debugging a new operating system on top of an old one, running on the same hardware. Legacy applications written for the previous system can continue to run, while applications are ported to the new system. Isolation is easier, since typically the VMM is much smaller than a traditional, full-service operating system – the VMM need only support the services provided by the raw hardware. And other services, such as checkpointing and migration, become much easier, since operations that block in the OS can be checkpointed by simply checkpointing the entire OS running on the VMM.

The design question is: how would you build a VMM for the Internet? Such a system would be able to host itself, and the Internet protocols, and potentially a completely different network design, all on an emulation of the physical network hardware. The goal of such a system would be to improve isolation and security: your VMM should be designed to provide strong isolation between virtual Internets.

There are two parts to the question, which you should consider separately. First, how would you virtualize the Internet from the perspective of end hosts? In other words, could you design a VMM that made it appear (to the OS and applications running on top of it) that the machine was connected to the Internet, when really it was connected to your new Internet VMM? Second, how would you virtualize Internet routers? Routers exchange packets with each other for both data (to forward packets to the destination) and control (to set up tables to know where to forward packets, to recover from failures, and so forth). How would you virtualize them as well?

For your Internet VMM, you may modify end host software or router hardware or software. You may not change the Internet protocols, however! We are much less interested in protocol specifics, than with the sketch of your general approach – e.g., try to keep the writeup to 1-2 pages.