CMPSCI 630 Systems **Spring 2022**

March 9 Lecture

Lecturer: Emery Berger *Scribe: Thomas Fang, Juelin Liu*

8.1 Research Centers Background

Historically, research centers were typically created by companies that monopolized a certain field. The motivation is to employ experts to keep up with cutting edge research and maintain that company's competitive advantage. Some prominent examples include:

- Bell Labs (AT&T)
  - Monopolized the phone industry at the time.
  - Bell Labs actually had research in other fields too. They had done some work with transistors and created UNIX.
- Paulo Alto Research Center (Xerox)
  - Monopolized photocopy industry.
  - Xerox had a computer called the Alto. They invented Mesa on the Alto. The Alto actually resembled a modern interface (which consists of mouse, keyboard, screen and GUI) compared to other computers at the time. Despite having these advanced technologies, Xerox didn't really plan to expand too much in the computer industry. Later on Bill Gates and Steve Jobs actually ran ahead with their own companies profiting from some of these ideas.
- Microsoft
  - 1975 Gates and Allen wrote a BASIC interpreter with FORTRAN syntax for the Altair (computer).
  - IBM monopolized PC industry. When the PC market blew up, Microsoft also became a big player since the IBM computers all ran MS-DOS and it charges the licence fee per machine.
  - Windows 95 was super popular during its time.
  - Research labs working for Microsoft introduced kernel architecture (Mach) for operating systems. It is often considered one of the earliest example of micro-kernel.

8.2 Microkernel vs. Monolithic Kernel

One of the original architectures for operating systems was the monolithic kernel. In the monolithic architecture, the entire OS is stored in the same kernel space. One advantage of this architecture is that the communication between components within the kernel is very fast. Since all components share the same space, data can be transferred through pointer. The main disadvantage is that it has lower fault tolerance and higher security risks. Since all the parts are in the same space, faulty software from one part (ex. driver) can crash the entire system. Since all the parts have the same privilege, any component can become a security risk.

The microkernel was introduced at CMU to address these shortcomings of the monolithic kernel. In the microkernel architecture, the different parts of the kernel are separated. For example, scheduler, pager, device driver, etc... all have their own space. The advantage is that there is isolation from bad software. If one driver goes down, that faulty software will not affect the other parts of the kernel. The downside is that communication can become slow and inefficient since it now requires message passing between different parts of the kernel.

8.3 Security

Security didn't always have the same emphasis it has today. Back in the day operating systems really weren't built with security in mind. For example, Windows used to pretty much let the user run anything they wanted. Unfortunately, it is much harder to retrofit security if the original design didn't include it.

The "Chicago problem":

The landline system back then is actually more reliable than landlines today which use IP and are susceptible to packet loss. However, the potential problem to this circuit based system is that if one crucial node fails (Chicago), the entire system collapses.

One way to reason about this is through the threat model. That is, when considering security measures, it is with respect to a particular threat. In the case of the Chicago problem, the threat would be taking over a substantial part of the US, which didn't really seem to be a serious hypothetical threat. In the case of computers, the idea of viruses seemed unrealistic at the time. Computers were not very mainstream, the idea of Worms and Trojan viruses seemed too complicated to be actually created by anybody.

The first major breach of computing security was the Morris Worm (1988). It leveraged vulnerabilities of each system and propagated through the network. This was also one of the earliest cases where a felony was assigned because of cybercrime. Thus, Robert Morris, who is an established scholar, is also a convicted felon due to his violation of 18 U.S.C. 1030(a)(5)(A).

Even in 1995, the Aleph One paper (Smashing The Stack For Fun And Profit) discussing vulnerability of stack (buffer overflow) was considered hypothetical and people did not think it would really happen. It wasn't until 2002 where Microsoft seriously implemented security measures with the Gates security freeze.

Examples of security and its development in different applications. This is another area where ontogeny recapitulates phylogeny often applies.

- Multics - has capabilities and permissions through access control list (ACL). Unix, has 3 layers of rwx permission control.
- iOS - employed large amount of isolation. For example, Notes, FB, mail are all users that are separated. The "chroot" command also let you apply OS level virtualization that changes what appears to be the root of a file system, hence isolating it.
- Android - adopted install time privileges which is almost no security. Next it employed "crowd favorite" unaudited app store which for obvious reason is not very secure either. Now it uses a sandbox approach which allows for isolated instances and hence more secure system.
- SQL injection attacks - not necessarily an application, but back then, this was a common security threat. Search programs which took input from the user to query the database were liable to "Select * from ;DROP TABLES" deleting the entire database.

- Memory access. Started with direct memory access (DMA) which is not secure. Next came remote DMA (RDMA) which involves socket loading from the input output memory management unit (IOMMU) which acts as a buffer.