CTIC
Center for Technology, Innovation and Competition

Penn Law
http://www.law.upenn.edu/academics/institutes/ctic/

University of Pennsylvania Law School
3400 Chestnut Street
Philadelphia, PA 19104-6204

Roundtable on Security for Cyber-Physical Systems*

One of the most important technological developments is the growing importance of cyber-physical systems (CPS). CPS are connected networks of computational nodes that interact closely with their physical environment. Unlike general purpose computers of the type connected to the Internet, the devices comprising CPS are more specialized. These networks can range from being loosely coordinated to tightly federated. CPS systems are also generally designed as platforms on which third parties can operate devices and software. Applications in CPS tend to be more safety-critical. Prominent examples of CPS include self-driving cars and medical device platforms.

One of the challenges is that CPS were not designed with any security. This roundtable is part of an NSF project focused on rectifying that shortcoming. Attacks on CPS can come both in the digital form and via the physical environment. Typically, each individual device is subject to greater resource constraints than a monolithic server environment. At the same time, each device must be more self-reliant in storing data and performing security, because other nodes in the network may be compromised or otherwise untrustworthy.

As a general matter, product designers note that it is impossible to provide perfect security. Designers can work to protect against known attacks. They face greater difficulty in protecting against novel or "zero day" attacks, some of which will inevitably succeed, since proactively eliminating every flaw is a practical impossibility. In addition, it is always possible to add more security. Designers are looking for help in framing the tradeoff for determining when to stop adding security. This requires a basic understanding of duties under negligence law as well as the standards for defective design—whether based in a risk-utility calculus, consumer expectations, or some other basis. Designers would also like to know more about the role of general industry practices and the adoption of formal standards, to determine how much effort they should invest in them.

The NSF project takes a novel approach to security. The major design shift is that security failures are acknowledged to be inevitable. The NSF project combines multiple layers of protection, including (1) prevention methods such as encryption, (2) fast detection of and recovery from failures, (3) fusion of diverse sensor technologies for robustness, and (4) tamper-proof data logging for forensic purposes.

(1) Prevention is the first line of defense. The NSF project incorporates context-sensitive encryption instead of full encryption for everything. CPS present different security requirements than general-purpose computers. CPS tend to be more sensitive to network denial-of-service attacks, so security must be fully embedded on each individual device—rather than relying on, say, an interactive key exchange. All encryption can be broken if given enough time and

* This work is supported in part by NSF CNS-1505799 and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy.

resources. CPS devices are also resource-constrained, so the algorithms must be more lightweight than what might otherwise be considered state of the art. Moreover, attackers will be able to gain physical access to the devices, so the encryption scheme must be able to perform additional functions such as protecting data integrity and preventing data forgery.

(2) Even with preventive measures in place, it is inevitable that some failures (including zero-day attacks) will still occur. The NSF project aims to guard against these unknowable threats by focusing on fast detection and fast recovery. Often, in CPS, a temporary failure is acceptable as long as the system detects the failure and recovers within a bounded amount of time. For example, an airplane retains enough forward momentum to remain in its flight envelope even if its engines fail for a short period of time. As long as the engines recover quickly enough, the flight is unaffected. The salient question is how large a window of time can be tolerated.

The NSF project is developing a method for guaranteeing time-bounds on both detection and recovery. The basic idea is to perform decentralized, peer-to-peer auditing, in which each node periodically validates that the current behavior of other nodes in the distributed network matches their expected behavior based on historical data. If the results for a given node do not match, then that node is flagged as faulty; and if a node is flagged by enough of its peers, then it is disconnected from the network until it can be fixed. The advantage of a detection/recovery approach is that it promises greater robustness against attacks, even unforeseen ones, because it does not depend on having to know anything about the specific manner of attack.

This method has some limitations. If too many of the nodes are corrupted (e.g., more than one third), peer auditing may not successfully identify when a node is corrupted. In addition, the system must be able to tolerate some bounded window of nonperformance. The shorter that time-bound guarantee must be, the more expensive it becomes in terms of computing resources. How to determine that cutoff is a major question.

(3) CPS also need to protect against physical attacks, not just cyber attacks. CPS interact with the physical world through sensors, which collect data from the outside environment, and actuators, which perform actions upon that environment. Because sensors and actuators are vulnerable to attacks and failures, CPS often fuse together a number and variety of different sensors and actuators. This redundancy and heterogeneity allows for greater robustness against overall system failure, but it also introduces greater technical challenges in terms of identifying when individual components have become corrupted. The NSF project utilizes state-estimation techniques to build a mathematical model of the system's expected sensor measurements. Attacks on sensors can be detected when the actual sensor measurements deviate significantly from the modeled estimates. Here, too, the degree of protection depends on a number of factors, including the complexity of the CPS, the amount of noise allowed in sensor measurements, and the allocation of constrained computing resources.

(4) The methods used by the NSF project offer an additional benefit in the form of tamper-proof forensic data logging, akin to black box crash recorders. Specifically, the detection and recovery methods are premised on the maintenance of lengthy data logs, which can serve the secondary function of providing an exact reconstruction of events occurring within the CPS.

Some unanswered questions include what data should be kept for legal purposes and for how long this data should be kept.

* * *

We have divided the roundtable discussion into four sessions, although we expect that much of the discussion will overlap. We have designated two scholars to kick off the discussion for each session.

The first session will begin with negligence law. The lead discussants are John Goldberg and Kyle Logue.

- What if any duties of care do manufacturers of CPS owe consumers and/or the general public?
- Should self-driving cars be held to the safety standards of ordinary cars, or to those of self-driving cars?
- What role would industry standards play?
- How should the reasonableness of burden be calculated with respect to software manufacturers as opposed to physical manufacturers?
- Does that calculus change if the software is considered to be a safety measure?
- What information should the CPS be designed to retain for forensic purposes in tort litigation?
- How do courts determine the scope of manufacturers' duty to protect against foreseeable misuse?
- What liability do manufacturers have for customers who do not use safety features (such as not using seatbelts)? Would turning off an autonomous driving feature be treated the same?
- What duties do manufacturers have to disclose and fix security vulnerabilities that are discovered post-sale? Does the lower cost of patching software affect the analysis?
- What role do warnings play?
- Is there a role for learned intermediaries?

The second session will focus on products liability law. The lead discussants are Mark Geistfeld and Robert Rabin.

- Is the standard for products liability determined by risk-utility or consumer expectations?
- If different states apply different products liability standards, how should product manufacturers manage their designs?
- Do those standards play out differently for software than for physical products?
- If the standard is risk-utility, is determining what constitutes a reasonable alternative design different for software, in which alternatives are potentially fairly cheap?
- How rigorously does software in CPS need to be tested?
- CPS are generally designed to operate within certain parameters. What liability do manufacturers have for extreme conditions that fall outside the design parameters?
- The length of time that the design tolerates failure can be shortened by committing additional resources to security. At what point does the software become "defective" for purposes of manufacturer liability?

- Similarly, the design can make fewer false negatives by committing more nodes to the peer review process. How should designers make this tradeoff?
- Does a duty of crashworthiness extend to CPS, particularly autonomous vehicles?
- Might artificial intelligence and autonomous behavior be inherently dangerous, akin to pharmaceutical drugs and medical devices?

The third session will turn to multiple causation and joint liability. The lead discussants are Donald Gifford and Michael Green.

- How should liability be apportioned as we shift from simple, self-contained products to complex, distributed platform systems?
- Product failure may arise from the interaction of multiple components that individually appear to be properly designed. Who should be held responsible for this "composition error"?
- Is the answer affected by the fact that the software community generally acknowledges the lack of tools to validate that a software system is error free?
- What might distinguish foreseeable software errors from unforeseeable ones?
- If an injury occurs because a hacker exploits a security hole that should have been patched, how should responsibility be apportioned?
- Should such hackers be considered an intervening cause?
- Is there a role for contributory negligence/misuse by end users or by third parties making use of the platform?
- Is there a role for concert of action, alternative, enterprise, or market share liability?
- How can we address these risks without shutting down the development and use of such platforms?

The final session will explore the potential role of federal regulation for CPS. Congress and federal agencies have a strong tradition of intervening on matters of public health and safety. It also has been keen to act on questions of Internet liability. The lead discussants are Catherine Sharkey and Benjamin Zipursky.

- What is the potential scope of federal preemption over state tort law in this area?
- How likely is FDA or NHTSA action to preempt security?
- In order to achieve preemption, would manufacturers have to obtain regulatory clearance of every single software upgrade or change? Are there other burdens that may make regulation and preemption a less attractive option than tort liability?
- Often, areas where preemption is recognized also seem to be accompanied by close agency oversight. Is that the price of admission, or is it possible to obtain federal preemption without heavy administrative process?
- What have been the different types of alternate compensation schemes enacted under tort preemption regimes? What are the relative pros and cons of these different setups?
- Even where preemption is recognized, tort lawyers sometimes find ways to win liability judgments anyway. What are some common workarounds to preemption?

Thank you again for participating. We look forward to what should be a stimulating discussion.