Justification for inclusion in NSPW of "Automatically Designing Consistent Deception Plans in Defense of Computer Systems"

This is a research paper addressing the idea of using automatic deliberate deception (lying and cheating) to defend computer systems. Rather than merely denying access to an attack, our paradigm is to deceive the attacker into either thinking their attack cannot succeed or thinking that is has succeeded when it has not. This could waste the attacker's time and effort more than merely denying access could, and delaying tactics can be critical to defending against a surprise attack. It would work as a second line of defense when access controls have been breached or in the case of insider attacks; multiple lines of defense are essential in an era of increasingly sophisticated and automated attacks on important computer systems. We propose deception for high-priority computer systems, such as those of a national infrastructure, whose continued operation is essential to society.

This is a new paradigm because people expect computer systems to be obedient servants to their commands. While new releases of complex software like Microsoft Windows often give incorrect error messages, they are not trying deliberately to lie (though some might argue). But lying and cheating serve justifiable functions in many human activities, including law, business, entertainment, politics, and military affairs. Deception is essential in these areas as a way to manipulate people for their own good or the good of society, or to prevent greater harms from occurring. Deception is well accepted in military science, for instance, as an essential component of mission planning, a way to take greater advantage of one's limited resources. It is reasonable to try to address all human activities with computer systems, so it is time to consider automation of lying and cheating.

This departs from current practice in software engineering in that most professionals in all fields follow codes of ethics in which lying and cheating are usually considered unacceptable. The reference (Bok, 1978) cited in the paper represents the usual ivory-tower academic analysis: Lying is unethical except in few rare circumstances. However, those circumstances usually include situations of possible serious harm without deception, and destruction of the software of a computer system by an attacker could be considered a serious harm, analogous to assault on a human being. Computer systems could thus be allowed recourse to deception in serious circumstances. But working out a policy on those circumstances, and deciding how to deceive, is a challenging problem worth investigation – hence our paper.

Deception in defense of computer systems was first introduced with honeypots, decoy computer systems used to collect attack data. But simple deceptions in file systems and resources are sufficient for most honeypots, since their goal is to *encourage* malicious behavior, with a few key exceptions like launching attacks on other sites. We want to defend ordinary systems with deception; this requires more-complex deceptions. The previous work we have published has analyzed the most effective place to interfere in a known attack plan, and has modeled the degree of suspicion an attacker had that they were being deceived, but this is our first work to directly address the notion of consistency in a deception using logical reasoning. It is also our first work to introduce decision trees to model the decisions that both attacker and defender must make.