# Harrison Nicholls

## Spring 2023

### CSCI383

What did I do? A close reading of *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments*: https://www.nsa.gov/portals/75/images/resources/everyone/digital-media-center/publications/research-papers/the-inevitability-of-failure-paper.pdf

### Summary

This is a research paper titled "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments" published by authors Peter A. Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, John F. Farrell (all affiliated with the NSA) in 2014. The paper argues that modern computing environments are inherently insecure due to the complexity of software systems, the difficulty of accurately identifying and mitigating vulnerabilities, and the constant evolution of threats. In particular, the authors point to the security mechanisms present in existing operating systems as a factor that makes security failures inevitable. It details 'mandatory security' and the notion of a 'trusted path', and contrasts them with what it claims are insufficient existing operating system features.

The paper also goes into concrete examples of current approaches to security in OS, and explains how the security of these approaches is insufficient.

The authors' motivation, they claim, is to build interest and start discussion in the area of OS security. They want to initiate a sort of cycle in which demand for secure operating systems leads to research in the area, eventually with the hope that commercially viable secure systems are created.

### Mandatory Security

The authors argue that the narrow definition of mandatory security provided by TCSEC (Department of Defense Trusted Computer System Evaluation Criteria) is insufficient for the needs of both the Department of Defense and private industry. Instead, they use a more general definition in which a mandatory security policy is any policy where the definition of policy logic and security attributes is controlled by a system security policy administrator. The authors also discuss the different types of mandatoryy security policies, including access control policies, authentication usage policies, and cryptographic usage policies. They explain that an operating system’s mandatory security policy may be divided into these categories. This notion of mandatory security seems sensible, as compartmentalization of policies simplifies the system, and makes security guarantees easier to analyze. This notion of analyzing security from different, separate perspectives is common practice in cryptography.

### Trusted Path

They then go on to elaborate on the notion of a 'trusted path.' A trusted path, they explain, is a mechanism that allows a user to interact directly with trusted software. It ensures that malicious software cannot impersonate trusted software or trick the user into believing that a function has been invoked without actually invoking it. The notion of trusted path can also be generalized to include a 'trusted channel' for communication between trusted software on *different* network devices. The authors explain that commercial operating systems, unfortunately, lack support for trusted path or protected path mechanisms. A notable exception is Microsoft Windows NT, which *does* provide a trusted path for login authentication and password changing.

### Access Control

The paper then gives general examples which motivate the need for mandatory security and trusted path notions. First, the authors discuss the problem of access control. An access control mechanism consists of enforcers and deciders. When someone tries to access an object, the enforcer invokes the decider component and provides it with some parameters. To prevent a malicious agent successfully tampering with any components, the authors argue that mandatory security mechanisms and a trusted/protected path mechanism are necessary. In particular, access control requires a trusted path mechanism to ensure that access is not 'propagated' without the user allowing it.

### Cryptography

Next, the authors argue that mandatory security and trusted paths are also necessary to address potential cryptographic vulnerabilities. They break down application-space crytpography into two components: invocation of mechanism, and mechanism. They claim that even where a hardware token implements cryptographic functions securely, an insecure operating system leaves them vulnerable. Malicious applications or users could still tamper with the hardware token, or they could be used by unauthorised parties. It's clear that the authors frame each of these issues with particular attention towards their two main talking points.

### Real-World Examples

The authors finish the body of the paper by discussing some concrete, real-world examples in which actual security solutions rely on the assumption of a secure operating system. Their point in doing this is to drive home the idea that trusted path and mandatory security notions are needed, and that in fact we might already be relying on their existence implicitly. This sounds quite dangerous, as they explained much earlier that operating systems already fall short of guaranteeing security in provable ways. In particular, they discuss mobile code, kerberos, IPSEC network security protocols at the IP layer, and firewalls. Having not known much about firewalls prior to reading the paper, I decided to spend extra time in understanding this section.

Network firewalls are used to create and enforce a boundary between two networks. They are security system that control incoming and outgoing network traffic, *typically* acting as a barrier between an internal network and the internet. They provide a range of security features, which are largely configurable by whoever sets up the firewall. Something I suppose is largely ignored, but that this paper poses as a serious vulnerability, is that firewalls do not offer protection from adversarial behavior on the part someone *inside* the network. Malicious parties on the inside can do harmful things like constructing 'tunnels' for outside malicious parties and leaking data. The authors claim here that malicious insiders leaking data can be addressed by mandatory security mechanisms in their host OS. The authors don't address the trusted path notion in this section, as presumably it doesn't do much to directly address firewall-related concerns.

## The need for a tapestry of security solutions

In the System Security section, the authors acknowledge that OS security features are not sufficient (although they clearly believe they are necessary) for what they refer to as 'total system security.' Application-specific security of course needs to rest on top of a secure OS to maintain this security. They even go so far as to admit that there are cases for which application specific security is *more* prudent than OS security. They offer the example of a digital signature scheme being implemented for an e-commerce application. They also mention that 'covert channels' are a persistent challenge for OS designers aiming for security, although they don't elaborate much on what a covert channel is.

## Closing points

I think that the content of this paper does a good job of building intuition and understanding for why OS security is so importan, albeit from a slightly 'historical' lens. I came away from it having learned a lot about the distinctions between OS-level and application-level security mechanisms, and feeling quite convinced of their narrative about trusted paths and mandatory security notions. It's also left me wondering just how much of this has been addressed since 1998.