Paul M. Washburn
CSCI E-11 Unit 3: IoT
December 18, 2019

Future Technology: Office Networking App – Cybersecurity

To keep Dr. Brown's data safe, this Office Networking App will employ elaborations of three defensive strategies: prevention, resilience under attack, and incident detection & recovery. A holistic view of security will be employed to implement a robust systems-wide security architecture so we can ensure we've taken all steps that are feasible and prudent towards the goals of protecting Dr. Brown's, as well as other users', data confidentiality, integrity and availability.

Given we won't know in advance who might attack, or what they might do, we need a threat model that is conservative yet realistic. Since cybersecurity is truly a property of our computer system as a whole, a holistic threat model would not only include software vulnerabilities. Our threat model will also take into account physical attacks & social engineering attacks. We will assume that an adversary that poses a threat to the Office Networking App's security will be capable of controlling some (not all) of the network's computers, controlling some (not all) of the software on these machines, as well as identifying & exploiting relevant bugs in our software base that may pose a security threat to our ecosystem. We assume an adversary will be capable of stealing secret keys, using public keys alongside them (and using them to reverse-engineer ciphertexts), and exploit system protocol. Furthermore, we'll also assume that adversaries might try to use social engineering attacks to gain access to individual or administrator user accounts via deception and/or outright theft.

There are numerous ways that we can *prevent* certain known attack vectors. Some standards such as the use of securely typed programming languages and employing public-private key cryptography will be bread-and-butter approaches, yet other prevention mechanisms will be instituted as well. To avoid code injection via string fields in the application, data sanitization will be employed for server-side tasks that typically concatenate strings from a user. In order to avoid confused deputies accidentally leaking data to unauthorized users, security labels will follow data to verify whether it can be exposed to other users via the privacy policy. (Resin on Rails will be the web framework we'll use to implement this feature.) Server-side, this app will consider a new app architecture that is Android-like in its concepts in that we will seek to implement isolation between applications & services (e.g. User databases will be separated from Payment databases, and all functionality will be isolated). Randomized disk-block encryption can also be used to avoid system overheads, keep file encryption design simple across the organization, and ensure the

same encryption system works on different file systems — keeping Dr. Brown's files safe even if an adversary with knowledge of our cryptography tries to plop down a different operating system during a breach. A Trusted Platform Module can be used to implement measured boot — preventing adversaries from accessing server applications without an exact *TPM_extend* match. The final prevention mechanism that we'll use server-side, which will help with the Trusted Platform Module, is Microsoft's BitLocker. On the user-side, this application will employ built-in biometric authentication mechanisms will be combined with signed boot procedures to augment the application's authentication protocol. Finally, we will protect web users by employing HTTPS (so we can encrypt communication with users like Dr. Brown) as well as identify "trusted" regions of the network through which our traffic can flow. Prevention of these common attack vectors will help Dr. Brown and other users of the app trust that what they tell the Office Networking App, as highlighted in the vignette, will remain confidential.

In order to remain *resilient under attack*, the Trusted Computing Base will be shrunk as much as possible. One obvious way of making progress towards this end would be deployment of tamper resistant hardware since only portions of such hardware are "trusted". Ascend is a good option for such hardware given that it obfuscates pin traffic from an adversary's view. Another mechanism that will help us to keep Dr. Brown's data safe is privilege separation, splitting functionality onto several machines or VMs, which will help compartmentalize breaches and thus isolate the damage that can be inflicted by any one unauthorized breach. Privilege separation will also help us to quarantine subcomponents & systems if and when they are penetrated, thus minimizing damages to Dr. Brown.

As for the *detection & recovery* strategy, it is important to understand that compromises will be inevitable; we'll need both proactive and reactive strategies to mitigate damages to users like Dr. Brown as well (as to the organization) if and when we are breached. For recovery, a tool like Retro will be on-hand to help disentangle legitimate changes made by authorized users like Dr. Brown and those made by an adversary. This will minimize the input necessary from Dr. Brown to recover her information should some of her data be affected by a breach. Assuming an adversary will eventually gain control of a subset of our system, performing system roll-backs & re-executions will need to be performed such that we preserve as much legitimate user data as possible. This will be vital to keeping Dr. Brown and other users happy with our services.

The fact that there exist many untrustworthy network operators and users renders the Internet a somewhat insecure medium, and taking a straightforward & holistic approach to cybersecurity will help raise barriers to attack while also remaining resilient when one occurs. While complete security is impossible we still must make a best effort to protect users like Dr. Brown.