What's the Difference Between Secure Comms and Secure Systems?

Electronic Design
Richard Kenner
Wed, 2015-10-21 11:55

In today's cyberspace, security issues arise in two contexts: during communication between two computing systems and within the computing system itself. In this article we'll contrast the types of threats that characterize the two environments and the measures that can be taken to address them.

Let's start by going back to the basic security principle: What's the threat we're protecting against? This is relevant for both physical and cyber security. Is a retail store in a high-crime or low-crime neighborhood? Is a particular measure aimed at stopping shoplifting, armed robbery, or employee theft?

Related

What Does It Mean To Secure the Internet of Things?

Windows 10, IoT, and Embedded Developers

Common Embedded Vulnerabilities, Part 1: Code Injection

nmunication security protects against three threats: the interruption of the communication; an unauthorized son reading the communication; and unauthorized data being sent or authorized data modified by unauthorized persons. Computer systems contain data, so the same three threats relate to that asset: protection against deletion; unauthorized access; and unauthorized modifications or additions.

Because a computing system also provides services, two additional types of threat arise. The first is "denial of service" (DoS), where an attacker interferes with (or prevents) legitimate use of that system. The other is "theft of services" and involves unauthorized use of a computing system. This was more of a concern when cloud computing was called "timesharing," but is still an issue today because it can cause DoS or, perhaps worse, allow the use of one computing system to attack another.

These threats correspond to the traditional challenges to a security system—protecting assets against attacks on their confidentiality, integrity, and availability.

Communication Security

From this brief analysis we can gain some insights into how to protect against the threats. Let's start with communications. Clearly, if either endpoint is insecure, so is the communication. But communication security requires more. Even if both endpoints are secure, the communication may not be if it can be intercepted. Unless we have physical security over the entire link (quite rare), there's no way to protect against the interruption or interception of the communication, and thus we must protect against the other threats using cryptographic methods.

Cryptography is an art that's been around for centuries and remains fundamentally the same, despite becoming much more mathematically sophisticated. It relies on modifying a message by using a "key" known to at least one of those communicating (to both in a "secure key" system and just one in a "public key" system), but not to an attacker.

To "break the code," one needs to find the key, either by deducing it from the encrypted data or using some other method. One such technique is to obtain it from an insecure endpoint, which was one of the concerns about the Heartbleed bug from last year. Another method, if the key itself is transmitted via a communication between the parties, involves exploiting an insecurity in that communication. In the case of a "public key" system, there's a third method: The public and private keys are related by some mathematical property, and it may be possible to use that knowledge to determine the private key.

To deduce the key from the encrypted data, two things must be true. First, there must be a way of recognizing a valid decryption. If what's being transmitted is random data, we have no hope. Indeed, we can often predict the nature of some of the data. The term "crib" for this type of data originated at Bletchley Park, and various techniques were often used to induce an adversary to transmit a known word; for example, by mining a harbor and looking for its name. But we must at least be able to recognize when we've been "successful."

Second, these attacks are based on the same key being used multiple times (either directly or modified in some way) in the encryption of a message. If a unique key is used for each component, even a trivial encryption technique, such as exclusive-OR'ing each piece of data with the key, is completely unbreakable. These are called "one time pad" techniques.

Cryptanalysis is the field of using these techniques to break codes; the goal of a secure cryptographic system is to prevent those techniques from being effective, which has been a multi-century battle. Unfortunately, mathematical proofs don't help much here. We can certainly prove the code implements the desired encryption algorithm, and we can often prove certain properties of that algorithm. However, what we most want to prove is that there's no algorithm to find the key used to encrypt a block of text.

there *is* such an algorithm, and a quite trivial one: Since we're able to determine when we've succeeded, all have to do is to try every possible key and we'll decrypt the message! So we certainly can't prove the absence of such an algorithm. What we really need is to prove we can't find the key within a certain amount of time, but that's a far more difficult problem because it relies on assumptions both about computer technology and within theoretical computer science (look up "Does P = NP?").

System Security

The security of computing systems is a different problem and, luckily, one where mathematical proof techniques not only help, but are essential. Computer security can be described in terms of two concepts. *Authentication* involves knowing who somebody is, and *authorization* encodes the permissions/restrictions for each user—what they're allowed to read and/or write and which resources they're allowed to use. Authentication is based either on something known (e.g., a password), something a user possesses (e.g., a fingerprint, iris scan, or device), or some combination. Authorizations are essentially a table that indicates what each user (including an anonymous, unauthenticated user) is allowed to do.

Securing a computer system means having a properly working authentication mechanism that can't be defeated and ensuring there's no way a user can perform an unauthorized action. In most computing systems, those tasks are distributed. It's mostly the kernel's responsibility for users logged in to the system, but it's all of the outward-facing servers, such as web servers and SSH servers, that have that responsibility for external users. A bug related to authentication or authorization in any of these components can allow a security breach.

Unlike communications security, proper functioning of authentication and authorization components in system security can be expressed in precise and formal mathematic language. It's also possible to prove that a component does perform those functions properly.

In fact, such a proof has been demonstrated. In 2002, the U.S. National Security Agency (NSA) funded a demonstration project for the Tokeneer ID Station, a software implementation of core security primitives that was proven by mathematical techniques to have the required security properties. Consisting of about 10,000 lines of SPARK code, the software is open source and available for download. Unfortunately, most systems aren't implemented using a core component for authentication and authorization, so this project is a proof of concept, not a mechanism that can currently be used to secure systems.

It's also important to note that the proof of certain security properties doesn't indicate the absence of bugs. Indeed, running AdaCore's CodePeer static-analysis tool, which didn't exist at the time, on Tokeneer along with additional code review discovered 11 bugs and code-quality issues, such as potential overflows and wrong variables being used. But we know from the proof that those bugs can't cause security breaches.

To summarize, communication- and computer-system security share some commonalities, mostly when we look at the threats we need to defend against. However, a lot of things that are different, mostly when we look at the mechanisms used to defend against those threats.

Source URL: http://electronicdesign.com/embedded/what-s-difference-between-secure-comms-and-secure-systems