
Deception is a powerful resilience tactic that provides observability into attack operations, deflects impact from production systems, and advises resilient system design. A lucid understanding of the goals, constraints, and design tradeoffs of deception systems could give leaders and engineers in software development, architecture, and operations a new tactic for building more resilient systems—and for bamboozling attackers.

Unfortunately, innovation in deception has languished for nearly a decade because of its exclusive ownership by information security specialists. Mimicry of individual system components remains the status-quo deception mechanism despite growing stale and unconvincing to attackers, who thrive on interconnections between components and expect to encounter systems. Consequently, attackers remain unchallenged and undeterred.

This wasted potential motivated our design of a new generation of deception systems, called deception environments. These are isolated replica environments containing complete, active systems that exist to attract, mislead, and observe attackers. By harnessing modern infrastructure and systems design expertise, software engineering teams can use deception tactics that are largely inaccessible to security specialists. To help software engineers and architects evaluate deception systems through the lens of systems design, we developed a set of design principles summarized as a pragmatic framework. This framework, called the FIC trilemma, captures the most important dimensions of designing deception systems: fidelity, isolation, and cost.

Why Embedded Software Development is Harder [3]

Developers sometimes distinguish "embedded" as a special kind of software development.
Maybe different than categories like "web" or "apps". Is it really? In the end, writing software looks very much the same.

There are aspects which have significant effect on the development process but it makes sense to differentiate six aspects.


Most of the platforms we use to communicate with each other online are tied to a single provider. But there's no technical reason for things to be this way. There are ways to chat online that offer a greater degree of privacy and freedom. Matrix is one of those ways.

[...]

The Matrix.org Foundation describes Matrix not as a protocol but as a decentralized conversation store. There is no single point of failure, such as a centralized server, in Matrix. When you communicate with someone, the data is shared among all the servers, with users participating in the conversation. If your server goes down, the conversation can still go on elsewhere until your server goes back up.

This is an effort to democratize communication. Each server has equal ownership over conversations and self-sovereignty over user data. These servers can be self-hosted at home or on a virtual private server or hosted by an organization, company, or community. This contrasts with all major commercial platforms, such as Discord, where the company owns the servers that everyone connects to and, as a result, all of the data. And if those servers go down, no one can communicate.

That's not where the contrasts stop. While someone on Twitter can't send a direct message to someone on Facebook, Matrix is interoperable by design. You can set up your Skype account, your Discord account, and your Slack account so that no matter where someone sends you a message, you see them in your Matrix client, and you can respond to any of them from one place. Matrix refers to this functionality as bridging.

**The QUIC API OpenSSL Will Not Provide** [5]

In a world that is now gradually adopting HTTP/3 (which, as you know, is implemented over QUIC), the problem with the missing API for QUIC is still a key problem.

There are a number of existing QUIC library implementation now since a few years back, and they are slowly maturing. The QUIC protocol became RFC 9000 and friends, but the most popular TLS libraries still don't provide the necessary APIs to make QUIC libraries possible to use them.
Source URL: http://www.tuxmachines.org/node/157405

Links: