Title
Fived: a service-based architecture implementation to innovate at the endpoints

Permalink
https://escholarship.org/uc/item/94z35496

Journal
ACM SIGCOMM Computer Communication Review, 40(4)

ISSN
0146-4833

ISBN
9781450302012

Authors
Capelis, DJ
Long, Darrell DE

Publication Date
2010-08-30

DOI
10.1145/1851182.1851240

Peer reviewed

Fived: A Service-Based Architecture Implementation to Innovate at the Endpoints

D.J. Capelis
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA
djcapelis@cs.ucsc.edu

Darrell D.E. Long
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA
darrell@cs.ucsc.edu

ABSTRACT

Security functions such as access control, encryption and authentication are typically left up to applications on the modern Internet. There is no unified system to implement these critical features. The access control that does exist on the network doesn't integrate well with user authentication systems, so access control decisions are based on the network location of a computer rather than the privilege level of its user. Just about every layer of the Internet provides optional encryption, yet most data on the Internet continues to be sent in the clear. Application developers routinely make mistakes in security critical code leading to bugs that manifest in worms, malware or provide a doorway for actively malicious attackers. We propose a unified session layer that integrates trustworthiness features into the core of the network. This would reverse the fortunes of security on the Internet and lead us toward a safer, more secure global network.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network communications; C.2.2 [**Network Protocols**]: Protocol architecture (OSI model)

General Terms

Design, Management, Security

1. INTRODUCTION

Our solution is a session layer that adds a set of core security services to the network. *Fived*, our prototype of this new session layer, provides services that put access control, authentication, encryption and other features directly into the network. Programmers of new applications can rely on our session layer instead of writing their own code to accomplish each of these tasks. Network administrators will have the tools they need to specify access in terms of which people can access what resources, not in terms of which addresses can send packets to what ports.

Our contribution is not in developing new novel networking features, but rather synthesizing many related concepts into a comprehensive, deployable and extendable architecture which paves the way for future innovations. Our session layer is designed for deployment and will serve as a boon for future Internet research, allowing designers to rapidly prototype and test new core services. Our session initiator design allows applications to move away from the notion of addresses and ports and towards the notion of two strings: one that specifies a host and one that specifies a service. Our system not only helps bring needed features to the Internet, but focuses on ensuring future researchers have the same opportunity.

2. DEPLOYABILITY

We present an architecture focused on *deployability* over cleanliness. Current estimates place the exhaustion of IPv4 addresses on the Internet as happening around the same time as SIGCOMM 2011 [1]. Over the next several years the Internet will be forced to come to terms with IPv6 and finally deploy the solutions networking researchers converged on 15 years ago. Unfortunately, this means any realistic deployment of new networking technologies reliant on commercial network operators to adopt new equipment, standards or practices will be delayed for some time.

Yet *fived* follows the end-to-end principle [3] and can be deployed on the edges of the network. Conservative core network operators do little to harm the adoption of our session layer. Another key component of a realistic plan is that the architecture must not require a large critical mass before organizations begin seeing benefits from the system. Users can use our architecture and begin enjoying its benefits as soon as they'd like. Simply downloading our software will allow them to start controlling access their internal services on their networks. We have designed multiple levels of compatibility software to ease transition for a diverse range of network environments. These comprehensive sets of compatibility libraries, layers and runtime tools provide the users the ability to obtain advantages from our architecture even before their network applications are adapted to interact with the session layer natively.

3. CORE SERVICES

The core services are the primitives we have selected for our the session layer prototype. These primitives can be used to embed the trustworthiness features into the Internet. We've written a prototype of the proposed session layer called *fived*. *Fived* currently runs in userspace and implements a portion of our core services. The session layer itself is loosely derived from the tcpmux protocol specified in 20 year old RFC 1078 [2]. The basic tcpmux protocol is simple and can be implemented in under 100 lines of C. Each of our core service extensions takes anywhere from tens of lines of

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.
ACM 978-1-4503-0201-2/10/08.

code to several hundred lines of code. These services work together to allow *fived* to provide a broad range of session services. The essential features include service multiplexing, role-based access control, transparent session-wide encryption, mobility, virtual hosting and distributed identities.

The following table provides a listing of the core primitives we've designed for *fived* during its initial deployment:

fived command	use(s)
LIST	service discovery
MULTIPLEX	multiple streams
AUTH	access control, auth
TLS	encryption
HOST	vhosting, VPN
GET/VERIFYAUTHKEY	distributed identity
ATTACH/DETACH	mobility

These primitives allow *fived* to provide functionality unavailable on the existing network as well as functionality currently provided by firewalls, VPNs, virtual hosting, zeroconf and other technologies. The solution *fived* provides over these others is integration into the session layer in the network. This constrasts with current solutions that integrate poorly or not at all with the network and make access decisions based on a computer's address rather than the user controlling it. *Fived* enables higher level policy decisions while requiring no hardware changes and remaining easy to deploy.

3.1 Multiplexing

One of the most critical and important commands in our architecture is the MULTIPLEX command. This allows a single session to be used for multiple streams of data and helps transform what would be rather primitive sessions into a robust and feature rich layer. The header format is as seen below:

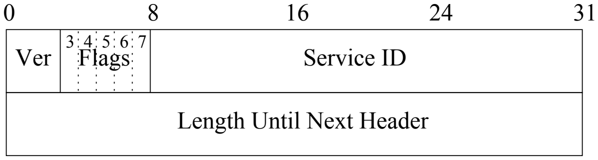

Figure 1: MULTIPLEX header format

The header allows *fived* to note a version number for future change, disambiguate between streams and pass along information so legacy applications can use existing semantics to run on *fived* sessions.

3.2 Session Initiator

A major component of our architecture addresses a larger goal of breaking network applications' dependence on the lower layers of the Internet to allow future researchers to innovate. While DNS allows applications to grow beyond addressing to naming, the networked application remains responsible for the name resolution and ends up bound to specific addressing semantics. The session initiator provides a solution: Instead of forcing applications to establish their own connections to session servers, the session architecture takes control earlier. The session initiator offers a new Application Programmers Interface (API) that takes care of initial connection establishment. This allows applications using the session stack to transcend existing APIs focused on addresses and port numbers and simply ask the networking stack for two names: The name of the organization or computer the application would like to communicate with and the name of the service the application would like to access. The session initiator does the rest.

Once applications move away from using addresses and port numbers, the underlying architecture of the Internet has more freedom to evolve. The session initiator will be the only thing that will need changes to allow computers to connect to each other in entirely novel ways. This layer will be much easier to change than IP or TCP is today and is a more appropriate abstraction and interface for networked applications on the future Internet.

4. BEYOND USERSPACE

While part of our goals include changing the interface to the network, which inherently demands changes in userspace APIs, not all of our design is tied to userspace. While the *fived* prototype is implemented in userspace and a userspace implementation of a network technology as critical for lowering the barrier of entry for early adopters, *fived* is relatively simple and could be either accelerated by, or implemented entirely in hardware. The future of *fived* may lie in a hybrid approach where the vast majority of session operations can be handled in hardware, with fallbacks to a software session server daemon to service less common operations which may require more dynamic responses. This matches well with the architecture of many large network devices in the field today.

5. CONCLUSION

With a session initiator that moves applications away from reliance on lower levels and a widely extensible session layer that can add a new core service to the Internet with as little as 10 lines of C code, *fived* serves as a catalyst for future innovation on the Internet. With an approach that provides a unified session layer that provides solutions for some of the Internet's most pressing problems and considerably enhances the network's security posture, *fived* provides needed and vital services all on its own. With an end-to-end solution implemented concisely in software, *fived* is an architecture that has the potential to become reality.

6. REFERENCES

[1] Hurricane Electric, "Hurricane electric ipv4 exhaustion counters." [Online]. Available: http://ipv6.he.net/statistics/

[2] M. Lotter. (1988, November) RFC 1078. [Online]. Available: http://www.rfc-editor.org/rfc/rfc1078.txt

[3] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, p. 288, 1984.