

Design of a Network Architecture with Inherent Data Path Security

Tilman Wolf

Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA
wolf@ecs.umass.edu

ABSTRACT

Next-generation Internet architectures require designs with inherent security guarantees. We present a network architecture that uses credentials to audit traffic in the data path, where defenses can be employed often more quickly and efficiently than on end-systems.

General Terms

Design, Security

Categories and Subject Descriptors

C2.1 [Computer-Communication Networks]: General—*Security and protection*

1. INTRODUCTION

The current Internet has been vastly successful in achieving global connectivity between a large number of different networks, devices, and users. This success is due to the openness of the system and the general “permit-by-default” design. A major shortcoming, however, is the difficulty of providing inherent security guarantees in the Internet. To provide authentication, confidentiality, integrity, and availability, a number of additions have been designed, developed, and deployed over last few decades. These approaches range from cryptographic operations on end-systems and routers (e.g., SSL and VPN tunnels) to dedicated traffic monitoring and access control (e.g., firewalls and intrusion detection systems) to defenses against denial of service (DoS) attacks (e.g., anomaly detection and rate limiting).

In some communications scenarios, high levels of verifiable security are essential (e.g., financial transactions, military communication, remote medical procedures, etc.). With the advent of network virtualization [1] and its likely deployment in the next-generation Internet testbed, it has become realistic to consider the clean-slate design of logical network that can provide high levels of security that operate on the same physical infrastructure as other networks. An important question is how to provide inherent security capabilities

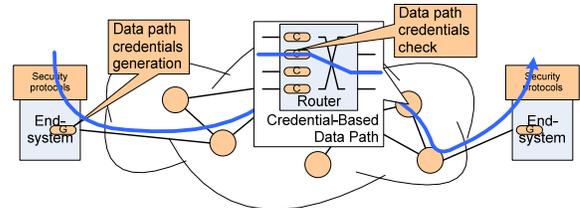


Figure 1: Credential-Based Data Path Architecture.

that do not simply rely on isolation (either physical or logical) that can be circumvented (e.g., insider attack).

We propose the design of a network architecture that provides intrinsic security guarantees and that is fundamentally different from the design principles of the current Internet. Instead of enabling communication access by default, we take the more conservative stance of deny-by-default. A similar idea has been vocalized by Ballani et al. [3]. In their work, it is proposed that routers do not forward packets unless the control path has negotiated such functionality. A system design to avoid denial of service attacks has been proposed by Yang et al. [4] where “capabilities” are used to identify legitimate traffic. It has been pointed out that capabilities are susceptible to denial of service attacks on the capability-granting subsystem [2]. In our architecture this problem can be isolated to affect only routers close to the source of DoS attack and thus limit the impact on the overall network. If a denial of service attack transmits packets with spoofed headers or without requesting proper credentials, it can be squelched within one hop of its source.

2. SECURE DATA PATH ARCHITECTURE

The network architecture that we propose is depicted in Figure 1. The key idea is to augment network traffic with credentials that can be audited in the data path. Each router performs a credential check and thus can positively identify traffic that is eligible for forwarding. Attack traffic with invalid credentials can be discarded. All routers along the data path of a connection participate in validating traffic and thus defending against attacks. In addition, end-system security protocols can provide orthogonal security features of confidentiality and integrity.

The system architecture of a router that implements data path credentials is shown in Figure 2 (conventional packet forwarding functions are not shown). In the control path, connections are managed and credentials are created. An end-system can request credentials for a particular flow.

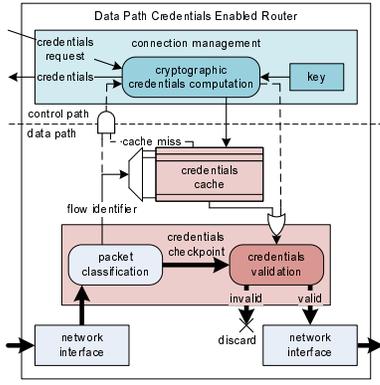


Figure 2: Router with Data Path Credentials.

These credentials are then computed based on the flow characteristics and the router’s cryptographic key. The resulting credentials are then transmitted back to the end-system and stored in the local credentials cache.

In the data path, packet headers are augmented to carry the credentials provided by the sending end-system. When a packet is received on the router for forwarding, the packet is first classified to identify to which flow it belongs. Then the credentials generated by the router are retrieved from the credentials cache. If the credentials match those in the packet, the packet is considered valid and thus forwarded. If the credentials do not match, the packet is discarded.

If credentials cannot be found in the local credentials cache, it may be due to the limited size of the cache or due to an invalid packet. It is possible to trigger a credentials recomputation (dashed lines in Figure 2) before discarding the packet. This process, however, may increase the systems vulnerability to denial of service (DoS) attacks since the cryptographic computation of credentials is an expensive operation. It is therefore important that the cache is sufficiently large and that new credential requests take priority over recomputations.

3. CREDENTIALS DESIGN

We propose a credentials design that is based on Bloom filters, which can maintain multiple credentials at the same time. The Bloom filter data structure superimposes credentials from each router along the path. The source node of a connection negotiates permission to transmit data across a router during connection setup. When router j , $1 \leq j \leq n$ permits transmission, it provides the source with router credentials r_j (computed from secret keys s_j and flow identified f). Router credentials are the set of indices $r_j[i]$, $1 \leq i \leq k$ of bits that are set in the Bloom filter array. The credentials from all routers along the path are then superimposed (i.e., logical OR operation) in the Bloom filter data structure. This creates aggregate credentials c (consisting of a single bit array of size m) that are sent with each data packet.

When receiving a packet with aggregate credentials c , router j can check the value of all bits that were provided in router credentials r_j . If the aggregate credentials are valid, then $\prod_i c[r_j[i]] = 1$, where the product is the equivalent of a logical AND operation. If the aggregate credentials do not contain the router credentials of a particular router, it is likely that one of the bits in credentials c does not contain

a 1 at one of the router credentials’ bit positions. Thus the validation of the aggregate credentials fails. This argument, of course, is of a probabilistic nature. A router may accept a packet that does not have correct credentials with the same probability as a false positive appears in the Bloom filter. However, packets are only successfully delivered to a destination if *all* routers let them pass. Thus, a packet with invalid credentials would need to encounter a false positive on every router along the path. This probability decreases geometrically with the number of hops in the path and thus is practically very small.

It is important to note that there exists a very simple attack to circumvent a credentials check: an attacker could set all bits in the credentials to 1. Such credentials would always match any router credentials. In order to make data path credentials immune to this attack, we introduce a “density” metric $d(c)$ that reflects the number of 1’s in credentials c as a fraction of the total size: $d(c) = \frac{1}{m} \sum_i c[i]$. To consider credentials valid, we require that the density is equal or below a certain threshold: $d(c) \leq d_{max}$.

Generating credentials based on cryptographic hash functions and flow identifiers ensures the following properties: (1) Data path credentials for different flows are different. (2) Data path credentials for flows that traverse different routers are different. (3) Data path credentials are difficult to fake since the result of the cryptographic hash function cannot be guessed without availability of keys s_j . (4) While the generation of credentials is computationally expensive, credential check operations are simple. (5) Data path credentials are of small and constant size. (6) Credentials cannot be “reversed” to obtain hash keys used by any of the routers or to create fake credentials.

With these key properties of data path credentials, it is possible to provide data path security features on the network architecture level.

4. SUMMARY

We have presented the basic concepts of an architecture for data path credentials that allow networks to closely control traffic. We have illustrated that credentials based on Bloom filter data structures can implement such an architecture and provide probabilistic guarantees that only permitted traffic can traverse the network.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. CNS-0447873 and CNS-0626690.

5. REFERENCES

- [1] ANDERSON, T., PETERSON, L., SHENKER, S., AND TURNER, J. Overcoming the Internet impasse through virtualization. *Computer* 38, 4 (Apr. 2005), 34–41.
- [2] ARGYRAKI, K., AND CHERITON, D. Network capabilities: The good, the bad and the ugly. In *Proc. of Fourth Workshop on Hot Topics in Networks (HotNets-IV)* (College Park, MD, Nov. 2005).
- [3] BALLANI, H., CHAWATHE, Y., RATNASAMY, S., ROSCOE, T., AND SHENKER, S. Off by default! In *Proc. of Fourth Workshop on Hot Topics in Networks (HotNets-IV)* (College Park, MD, Nov. 2005).
- [4] YANG, X., WETHERALL, D., AND ANDERSON, T. A DoS-limiting network architecture. *SIGCOMM Computer Communication Review* 35, 4 (2005), 241–252.