NetQuery: A Knowledge Plane for Reasoning about Network Properties

Alan Shieh Emin Gün Sirer Fred B. Schneider
{ashieh, egs, fbs}@cs.cornell.edu
Department of Computer Science, Cornell University

1. Introduction

Depending on their configuration, administration, and provisioning, networks provide drastically different features. For instance, some networks provide little failure resilience while others provision failover capacity and deploy middleboxes to protect against denial of service attacks [1, 2]. Yet the standard IP interface masks these differences; every network appears to provide the same basic "dial-tone" service. Consequently, clients that desire certain network properties must resort to ad hoc techniques to detect these differences or must target the lowest common denominator service.

This abstract outlines NetQuery, a general-purpose *knowledge plane* that enables reasoning about the network [3]. The knowledge plane distributes information about all physical network devices and logical entities participating in the network, enabling applications to determine whether the network satisfies their requirements.

Using such properties to reason about the network only makes sense when there is a basis for trusting the properties. NetQuery extends the execution platforms of network devices and their knowledge-, control-, and data-planes to provide this basis for trust, manifested as trustworthy properties that encode local guarantees on device and link behavior. Applications use logical analysis to combine such local guarantees, inductively forming network-wide guarantees.

NetQuery targets an environment where all networked components are assumed to be equipped with secure hardware coprocessors, though we provide a transition model as well. Coprocessors such as the Trusted Platform Module (TPM) are cheap, ubiquitous, and suitable as a root of trust for claims [4]. Such secure hardware makes it possible to issue unforgeable certificates describing the software environment and dynamic state of a device through *attestation* [5]. Attestation forms the basis for trusting remote devices to self-report their properties.

ACM CoNEXT Student Workshop, November 30, Philadelphia, USA.

Network management systems, widely deployed in networks, maintain representations of the network that are sufficiently rich for analysis. However, they provide few mechanisms for establishing the verity of facts and consequent remote reasoning. Malicious parties that might benefit from misleading applications could manipulate the representation of the network to change the result of analysis. NetQuery extends the data model of these network management systems to support inter-domain analysis that provides strong guarantees.

Securing the devices and knowledge plane are not enough. An attacker can still manipulate the knowledge plane by controlling the input to a trustworthy device, tricking it into reporting false information. NetQuery protects the control-plane and data-plane with novel protocols that prevent such attacks. These protocols derive their protection from cryptographic primitives that are already widely offloaded on network devices and hence add little overhead.

These sample scenarios illustrate applications that NetQuery enables:

Differentiating providers and peers. Critical applications can have stringent network reliability and performance requirements. Users of such applications currently differentiate between ISPs based solely on reputation and monitoring. By having devices self-report their properties, NetQuery enables applications to make informed decisions among competing ISPs. For instance, customers can verify that their VPN providers have provisioned enough MPLS backup paths to satisfy SLAs and ISPs can verify that their peers have reliable topologies; such checks are tractable assuming a bounded number of failures. While current contracts and service selection are limited to ad hoc reasoning about reputation and economic penalties, NetQuery enables mechanical a priori reasoning about failures.

Authentication of hosts and networks. Hosts can use NetQuery to check for insecure networks, and vice versa. For instance, wireless networks are currently identified by SSIDs, which attackers can easily forge to launch man-in-the-middle attacks. NetQuery enables clients to determine that a wireless network is trustworthy based on its structure (e.g., an access point from a trusted hardware vendor, connected directly to a trusted ISP). By comparison, existing PKI- and reputation-based approaches favors large incumbents that can af-

ford the requisite auditing and marketing costs.

2. The NetQuery knowledge plane

The knowledge plane maintains a representation of the network topology and configuration, decomposed into a collection of *factoids*. Factoids self-reported by devices typically encode low-level properties such as routing tables and neighbor tables. Factoids are signed to protect against tampering and to attribute them to their originator. Attribution enables consumers of factoids to reason about the trustworthiness of facts in the knowledge plane: an application can choose to use facts only from sources that it trusts. To protect against exposure of confidential information, each factoid is protected with an access policy. As described below, NetQuery provides sanitizers that control the release of this information while enabling analysis.

Existing management systems such as SNMP provide standard schemas for representing devices, such as routers and switches; and network abstractions, such as TCP connections and endpoints. These schemas together enable a comprehensive description of the hardware and software configuration of most ISP and enterprise networks; adapting SNMP to a NetQuery knowledge plane enables analysis to cover such networks. Internet applications may need guarantees that span multiple ASes; to support these, one can extend a federated management system such as RPSL to a knowledge plane.

Though TPMs, once deployed, provide a scalable root of trust that enables factoids to be attributed to devices, NetQuery facilitates incremental deployment by allowing factoids to be backed by other types of certificates. For instance, large access providers might be trusted due to reputation or can amortize the cost of a network audit. Smaller providers can opt to locally deploy of devices with TPMs. For instance, a wireless hotspot can advertise network quality to customers by combining such devices with an ISP-issued factoid certifying the access link capacity.

2.1 Using trustworthy computing

NetQuery uses the attestation primitive of TPMs to establish the accountability of factoids inserted into the knowledge plane. A TPM attestation is a remotely-verifiable, unforgeable, and tamperproof certificate that binds a software-specified message to the particular hardware and software platform that generated it [5].

Attestation provides the foundation for trusting devices. An attestation issued by a TPM on a secure router includes a description of the router's platform hardware, boot code, and OS, backed by a signature chain going back to the router manufacturer. Using this, a NetQuery client can determine that the router runs software and hardware revisions that are known to conform to IETF standards for control- and data-plane behavior and to export accurate factoids.

Forging an attestation requires access to the root key or hardware attacks. Attackers can induce non-compliant behavior if a device contains exploitable bugs; since such exploits can enable malicious attackers to compromise a network, equipment vendors and rational network operators have strong incentives to close these bugs. Since attestations encode version information, a NetQuery client can choose not to trust factoids from devices that are not patched against known exploits.

Data sanitization

NetQuery uses attestation not only for establishing accountability for factoids inserted into the knowledge plane, but also to expand the optimizations and abstractions that the knowledge plane supports. An operator can deploy sanitizers to protect the confidentiality of knowledge plane information. Rather than exporting factoids directly to external clients, sanitizers run analysis programs on the clients' behalf. Attestations ensure that analysis programs are running an acceptable software version on a suitable platform. Such services can be incrementally deployed, since they require only a commodity, TPM-equipped server.

Establishing link guarantees

TPMs only serve to establish trust about the devices and provide no direct assurances about the links. For instance, a dishonest ISP could use network virtualization to trick NetQuery routers into reporting a network that appears highly redundant, yet is built from a non-redundant physical topology (i.e., showing off a Potemkin-village).

Data plane enhancements can provide such assurances about privacy, integrity, and performance. Attestation of devices establishes that these enhancements are in effect. Deploying link-layer encryption, standard on data-center and carrier-grade devices, provides confidentiality assurances at line rate. NetQuery extends this functionality with monitoring-based performance assurances. This monitoring, which is cryptographically-protected against manipulation, verifies the purported capacity of links and effectively imposes mandatory admission control on any hidden layers.

3. References

[1] AT&T. AT&T Internet Protect Service, Aug. 2009. Available at `http://www.corp.att.com/abs/serviceguide/docs/ip\_sg.doc`.

[2] M. Casado, P. Cao, A. Akella, and N. Provos. Flow-Cookies: Using Bandwidth Amplification to Defend Against DDoS Flooding Attacks. In *IEEE IWQoS*, June 2006.

[3] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski. A Knowledge Plane for the Internet. In *ACM SIGCOMM*, Aug. 2003.

[4] P. England, B. Lampson, J. Manferdelli, M. Peinado, and B. Willman. A Trusted Open Platform. *Computer*, 36(7):55–62, 2003.

[5] M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson. The Digital Distributed System Security Architecture. In *Proc. 12th NIST-NCSC National Computer Security Conference*, pages 305–319, 1989.