A Taxonomy of Capabilities Based DDoS Defense Architectures

Vamsi Kambhampati
vamsi@cs.colostate.edu

Daniel Massey
massey@cs.colostate.edu

Christos Papadopoulos
christos@cs.colostate.edu

Dept. of Computer Science, Colorado State University
Fort Collins, CO 80523-1873

ABSTRACT

A recent class of DDoS defenses based on *network capabilities* advocate fundamental changes to the Internet. However, despite the many point solutions, there has not been a rigorous study of the entire solution space for capability architectures. We believe these changes will inevitably introduce engineering tradeoffs in effectiveness and deployability. To understand the tradeoffs, and identify challenges, we build a *taxonomy* to categorize possible options and map the potential solution space. Our taxonomy identifies key components of a capability architecture, separates implementation from fundamentals, and opens up directions to explore for building a capabilities enabled Internet.

Categories and Subject Descriptors

C.2.6 [**Computer Communication Networks**]: Internetworking; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design

General Terms

Security, Design

1. INTRODUCTION

Distributed Denial of Service (DDoS) attacks are a threat to the Internet. The open nature of the Internet allows any source to send traffic to any destination, *without consent from the destination*. A recent class of DDoS defenses based on *network capabilities* [4, 5, 3, 2] advocate fundamental changes to the Internet, so that senders must obtain explicit authorization (i.e., a *capability*) from the receiver before they are allowed to send significant amount of traffic.

Figure 1 shows a simplified example of how capabilities work. A sender that wants to communicate with a receiver, sends an initial *request* packet to the receiver. Routers on the forwarding path insert *pre-capabilities* into these requests. Upon receiving the request, the receiver synthesizes a *host-capability* from pre-capabilities and returns it to the sender. Capabilities use cryptographic techniques so that routers can easily verify their validity. Subsequent data packets from the sender must carry capabilities; otherwise routers would drop unauthorized packets. The receiver now has the ability to reject senders simply by not returning capabilities. Moreover, capabilities typically expire after a while, enabling the receiver to reject senders that misbehave after the initial communication.

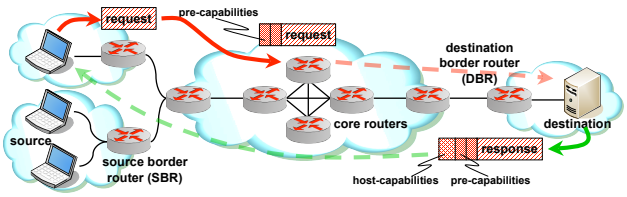

Figure 1: A network with capabilities.

Using capabilities, the destination can reject packets that it does not want to receive; for example, DDoS traffic. However, despite the fact that many point solutions have been proposed [4, 5, 3, 2], there has not been a rigorous study of the entire solution space for capabilities architectures. We aim to remedy this problem by re-examining capabilities from ground up. We believe adding capabilities to the Internet introduces fundamental challenges and engineering trade-offs in terms of effectiveness and deployability. To better understand these tradeoffs, we begin by building a taxonomy to categorize possible options and map the potential solution space.

Our taxonomy identifies key components that make up a capability architecture, sets apart implementation specifics from fundamental requirements, and opens up future directions to explore for building a capabilities enabled Internet.

2. TAXONOMY

Figure 1 shows a simple network model which outlines the different components of a network that utilizes capabilities. We identify five locations, the *source* and the *destination*, the source border router (SBR) and the destination border router (DBR), which act as administrative boundaries between the end-hosts and rest of the network, and *core* routers, which transit traffic for the edges. These locations are managed by different administrative authorities, and have different policies.

We build the taxonomy by first examining what capabilities the destination requires, and how they can be expressed.

Copyright is held by the author/owner(s).
SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

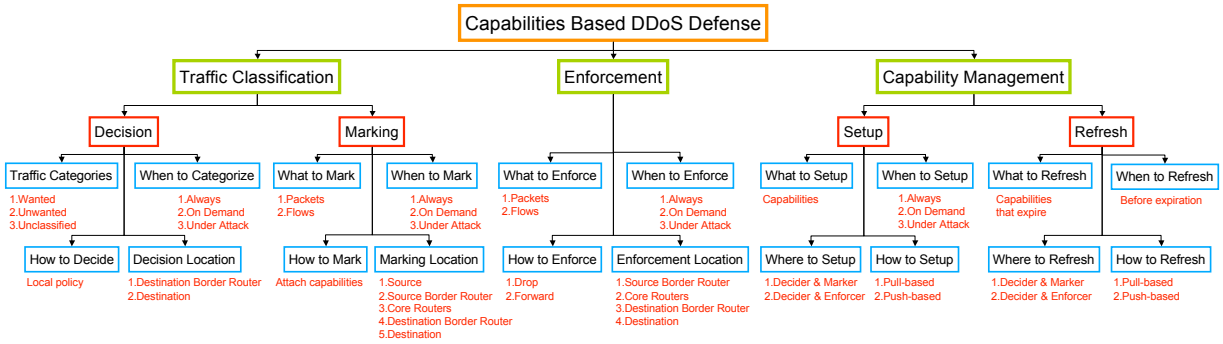

Figure 2: Taxonomy of capability architectures

Then, we try to understand how the network can enforce capabilities. Finally, we try to understand the communication between the destination and rest of the network, to exchange capabilities.

These steps lead to three top level categories in our taxonomy. First, **Traffic Classification** organizes packets into different traffic categories. Specifically it includes *Decision*, which determines the class of packets that should be given capabilities, and *Marking*, which assigns capabilities to the packets. Second, **Enforcement** implements the traffic policies specified by the destination, and finally, **Capability Management** deals with the process of establishing capabilities. Capabilities are managed in two different phases: *Setup*, which is responsible for granting initial capabilities, and *Maintenance* refreshes capabilities before they expire.

Although different architectures may choose different organizations, we believe there are three fundamental traffic categories: 1) **wanted**, 2) **unwanted**, and 3) **unclassified**. Fundamentally, capability architectures need to distinguish between wanted traffic, which is the traffic the destination wants to receive, and unwanted traffic, which may be dropped. However, there are times when the destination does not have sufficient information to categorize traffic (for example, first packet from a new client) or, it may choose not to categorize traffic (when there is no DDoS attack). In both these cases, the destination leaves the traffic as unclassified.

Figure 2 shows the taxonomy we developed. In general, our taxonomy asks important questions for each of these categories, such as: *What*, *Where*, *How* and *When* to decide, manage and enforce. The top level categories are the fundamental *actions* that need to be performed by a capability architecture.

We also seek to identify the key *actors* that implement the actions outlined above. Our taxonomy defines three actors: 1) the **decider**, 2) the **marker**, and 3) the **enforcer**, which are responsible for decision, marking and enforcement. Together, these three actors implement the capability architecture.

3. FUTURE DIRECTIONS

The taxonomy shows some clear directions we believe are important to be addressed by future capability architecture designs. In particular, one direction involves understanding where to deploy the decider, marker and enforcer. From our network model, these could be deployed at some combination of source, destination, *SBR*, *DBR*, or core, with different choices leading to differing levels of effectiveness and deployability. Most existing proposals do not throughly investigate the effectiveness of partial deployment. We believe there are alternate partial deployment strategies that may not substantially impact effectiveness. For example, instead of marking and deciding at the source and destination, these tasks can be performed at the border routers (i.e., *SBR* and *DBR*).

Another direction involves the question of how to bootstrap capabilities. Allowing a sender to request capabilities, as is done in most proposals, leads to *Denial of Capability* (DoC) attacks [1]. There have been some attempts at point solutions [2, 5, 3] to address this problem, however, as our taxonomy shows there are alternate approaches that have yet to be explored. For example, the decider may *push* a few initial capabilities to markers, with consideration to scalability (pushing capabilities to all sources is difficult to scale). A possible approach is to pre-distribute capabilities to intermediaries, which are closer to the sources and may use authentication. If no authentication is involved, the intermediaries may still confine the DoC problem to the local network. Exploring these alternate strategies for bootstrapping capabilities is important, particularly to address DoC attacks.

Apart from the above directions, we believe there are other important issues to consider in future capability architecture designs. For example, we need to understand how capabilities operate if end-to-end communications were to use multiple paths, or in a mobile environment where the source or the destination may be constantly changing location. These questions directly effect the deployment of deciders, markers and enforcers, or may have a direct impact on the architecture itself.

4. REFERENCES

[1] K. Argyraki and D. Cheriton. Network capabilities: the good, the bad and the ugly. In *HotNets-IV*, 2005.

[2] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu. Portcullis: Protecting connection setup from Denial-of-Capability attacks. In *SIGCOMM*, 2007.

[3] L. Wang, Q. Wu, and D. D. Luong. Engaging edge networks in preventing and mitigating undesirable network traffic. In *NPSEC*, 2007.

[4] A. Yaar, A. Perrig, and D. Song. SIFF: A stateless internet flow filter to mitigate DDoS flooding attacks. In *IEEE SSP*, 2004.

[5] X. Yang, D. Wetherall, and T. Anderson. A DoS-limiting network architecture. In *SIGCOMM*, 2005.