Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 14, 2015

J. Hildebrand
Cisco Systems, Inc.
P. McManus
Mozilla

November 10, 2014

Erosion of the moral authority of transparent middleboxes
draft-hildebrand-middlebox-erosion-01

Abstract

Many middleboxes on the Internet attempt to add value to the connections that traverse that point on the network. Problems in their implementations erode the moral authority that otherwise might accrue to the legitimate value that they add.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 14, 2015.

Copyright Notice

1. Introduction

There are several middlebox use cases that typically stand in the way of better encryption helping to mitigate perpass-style attacks.

- Local caching
- Enterprise policy controls, including Data Loss Prevention (DLP) and monitoring for acceptable use
- Service provider acceleration of mobile data
- Network management and quality of service routing
- Authorization and billing of network services

These use cases may cause third parties to an otherwise end-to-end conversation to have legitimate legal and moral rights that grant them participation in the conversation. This document discusses several reasons why the legitimacy of these use cases is undermined in the minds of some who build other products for the Internet.

2. Similarity to attacks

Some middlebox capabilities are currently implemented using the same mechanisms employed by attackers, including passive capturing of plaintext data, active impersonation, and denial of service. Further, some services are legitimate in one context but illegitimate in another - and the transparent nature of the middleboxes creates security problems separating those problem domains.

It is difficult to design protocols that simultaneously prevent a given vulnerability and simultaneously selectively allow legitimate access, and arguments that particular attacks cannot therefore be mitigated are greeted by end-users with skepticism - particularly when the benefit added by the middlebox does not accrue directly to those users.

3. Unintentional breakage

The experiences of living with a wide variety of middleboxes in the real world lead developers to realize that they all have defects that go years without being addressed. Even when the vendor fixes a given bug, software is updated so infrequently at this layer that often the bug must just be worked around.

Developers that have to add multiple special cases to their products as they discover every new way to incorrectly implement what they

previously thought were simple protocols often overreact by using protocols that are harder to manage, have worse security properties, or perform poorly.

Even middleboxes that are operating correctly become design constraints that inhibit end to end innovation because of their centralized model. A middlebox that inserts itself into all web traffic on a network but only speaks HTTP/1.1 will not allow the evolution of any device on that network beyond that state.

4. Support cost appropriation

When a middlebox subtly fails, end users never call the entity that deployed the middlebox, much less the vendor that built that box. Indeed, the nature of a transparent middlebox makes it very difficult to even diagnose the error for a professional. Instead, they file a support request with the services that they are trying to access.

The team that developed that service typically spends many hours finally tracking down the issue, only to finally find the problem with the middlebox. The original end user never has the authority to fix the middlebox or even opt out of using it. Instead they demand the service owner work around the problem. The service implementor may not have any more control than the end user, so too often the result is that new technologies have to be abandoned because they are not backwards compatible with middlebox infrastructure that neither the end user nor service operator has direct control over. This dynamic holds back Internet evolution.

When the costs associated with broken behavior are not paid by the developers of that behavior, it is easy for those developers to assume that everyone is happy with their product.

5. Other monetary incentives

Developers of new services will often try to make their network traffic as similar as possible to an existing essential service. This approach maximizes the chances that they will be able to develop a user base, however it can stress middleboxes beyond their design constraints causing them to fail in new ways.

When middlebox developers bring about their own downfall by pushing application providers outside of natural design patterns, they do not impress the community with their desire to be trustable elements of the Internet architecture.

6. Conclusions

When the moral authority of middleboxes is eroded, arguments by their developers to allow unfettered access to the plaintext of traffic that traverses those boxes may be called into question.

As an industry, we should look for other mechanisms to provide legitimate third-party value. Explicitly addressed intermediaries offer an alternative to transparent middleboxes. Addressing the harder problems of service discovery and authorization would make these services more effective, robust, and secure than their existing middlebox counterparts.

7. References

Authors' Addresses

Joe Hildebrand
Cisco Systems, Inc.

Email: jhildebr@cisco.com

Patrick McManus
Mozilla

Email: pmcmanus@mozilla.com