Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2013

R. Barnes
BBN Technologies
A. Cooper
CDT
O. Kolkman
NLnet Labs
October 22, 2012

Technical Considerations for Internet Service Filtering
draft-iab-filtering-considerations-01.txt

Abstract

The Internet is structured to be an open communications medium. This openness is one of the key underpinnings of Internet innovation, but it can also allow communications that may be viewed as undesirable by certain parties. Thus, as the Internet has grown, so have mechanisms to limit the extent and impact of abusive or allegedly illegal communications. Recently, there has been an increasing emphasis on "blocking" or "filtering," the active prevention of abusive or allegedly illegal communications. This document examines several technical approaches to Internet content blocking in terms of their alignment with the overall Internet architecture. In general, the approach to content filtering that is most coherent with the Internet architecture is to inform endpoints about potentially undesirable services, so that the communicants can avoid engaging in abusive or illegal communications.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The original design goal of the Internet was to enable communications between hosts. As this goal was met and people started using the Internet to communicate, however, it became apparent that some hosts were engaging in arguably undesirable communications. The most famous early example of undesirable communications was the Morris worm, which used the Internet to infect many hosts in 1988. As the Internet has evolved into a rich communications medium, so have mechanisms to restrict undesirable communications.

Efforts to restrict or deny access to Internet resources have evolved over time. As noted in [RFC4084], some Internet service providers impose restrictions on which applications their customers may use and which traffic they allow on their networks. These restrictions are often imposed with customer consent, where customers may be enterprises or individuals. Increasingly, however, both governmental and private sector entities are seeking to block access to certain content, traffic, or communications without the knowledge or agreement of affected users. Where these entities do not directly control networks, they aim to make use of intermediary systems to effectuate the blocking.

Entities may seek to block Internet content for a diversity of reasons, including defending against security threats, restricting access to content thought to be objectionable, and preventing illegal activity. While blocking remains highly contentious in many cases, the desire to restrict access to content will likely continue to exist.

This document aims to clarify the technical implications and trade-offs of various blocking strategies and to identify the potential for different strategies to come into conflict with the Internet's architecture or cause harmful side effects ("collateral damage"). The strategies broadly fall into three categories:

1. Control by intermediaries (intermediary-based filtering)

2. Manipulation of authoritative data (server-based filtering)

3. Reputation and authentication systems (endpoint-based filtering)

Examples of blocking or attempted blocking using the DNS, HTTP proxies, domain name seizures, spam filters, and RPKI manipulation are used to illustrate each category's properties.

Whether particular forms of blocking are lawful in particular jurisdictions raises complicated legal questions that are outside the scope of this document.

2. Architectural Principles

To understand the implications of different blocking strategies, it is important to understand the key principles that have informed the design of the Internet. While much of this ground has been well trod before, this section highlights four architectural principles that have a direct impact on the viability of content blocking: end-to-end connectivity and "transparency", layering, distribution and mobility, and locality and autonomy.

2.1. End-to-End Connectivity and "Transparency"

The end-to-end principle is "the core architectural guideline of the Internet" [RFC3724]. Adherence to the principle of vesting endpoints with the functionality to accomplish end-to-end tasks results in a "transparent" network in which packets are not filtered or transformed en route [RFC2775]. This transparency in turn is a key requirement for providing end-to-end security features on the network. Modern security mechanisms that rely on trusted hosts communicating via a secure channel without intermediary interference enable the network to support e-commerce, confidential communication, and other similar uses.

The end-to-end principle is fundamental for Internet security, and the foundation on which Internet security protocols are built. Protocols such as TLS and IPsec [RFC5246][RFC4301] are designed to ensure that each endpoint of the communication knows the identity of the other endpoint, and that only the endpoints of the communication can access the secured contents of the communication. For example, when a user connects to a bank’s web site, TLS ensures that the user’s banking information is communicated to the bank and nobody else.

Some blocking strategies require intermediaries to insert themselves within the end-to-end communications path, potentially breaking security properties of Internet protocols. In these cases it can be difficult or impossible for endpoints to distinguish between attackers and the entities conducting blocking.

A similar notion to the end-to-end principle is the notion of "transparency," that is, the idea that the network should provide a generic connectivity service between endpoints, with minimal interaction by intermediaries aside from routing packets from source to destination. In "Reflections on Internet Transparency" [RFC4924], the IAB assessed the relevance of this principle and concluded that "far from having lessened in relevance, technical implications of intentionally or inadvertently impeding network transparency play a critical role in the Internet's ability to support innovation and global communication."

2.2. Layering

Internet applications are built out of a collection of loosely-coupled components or "layers." Different layers serve different purposes, such as routing, transport, and naming (see [RFC1122], especially Section 1.1.3). The functions at these layers are developed autonomously and almost always operated by different entities. For example, in many networks, physical and link-layer connectivity is provided by an "access provider", while IP routing is performed by an "Internet service provider" -- and application-layer services are provided by a completely separate entity (e.g., a web server). Upper-layer protocols and applications rely on combinations of lower-layer functions in order to work. As a consequence of the end-to-end principle, functionality at higher layers tends to be more specialized, so that many different specialized applications can make use of the same generic underlying network functions.

As a result of this structure, actions taken at one layer can affect functionality or applications at higher layers. For example, manipulating routing or naming functions to restrict access to a narrow set of resources via specific applications will likely affect all applications that depend on those functions.

In a similar manner, the physical distance between a host and a service provider at a particular layer grows as one moves up the stack. A host must be physically connected to a link-layer access provider network, and its distance from its ISP is limited by the length of the link that connects it, but Internet applications can be delivered to the host from anywhere in the world.

Thus, as one considers changes at each layer of the stack, changes at higher layers become more specific in terms of application, but more broad in terms of the size and geography of hosts impacted. Changes to an access network will only affect a relatively small, well-defined set of users (namely, those connected to the access network), but can affect all applications for those users. Changes to an application service can affect users across the entire Internet, but only for that specific application.

2.3. Distribution and Mobility

The Internet is designed as a distributed system both geographically and topologically. Resources can be made globally accessible regardless of their physical location or connectivity providers used. Resources are also highly mobile -- moving content from one physical or logical address to another can often be easily accomplished.

This distribution and mobility underlies a large part of the resiliency of the Internet. Internet routing can survive major outages such as cuts in undersea fibers because the distributed routing system of the Internet allows individual networks to collaborate to route traffic. Application services are commonly protected using distributed servers. For example, even though the 2010 earthquake in Haiti destroyed almost all of the Internet infrastructure in the country, the Haitian top-level domain name (.ht) had no interruption in service because it was also accessible via servers in the United States, Canada, and France.

Undesirable communications also benefit from this resiliency -- resources that are blocked or restricted in one part of the Internet can be reconstituted in another part of the Internet. If a web site is prevented from using a domain name or set of IP addresses, the web site can simply move to another domain name or network.

2.4. Locality and Autonomy

The basic unit of Internet routing is an "Autonomous System" -- a network that manages its own routing internally. The concept of autonomy is present in many aspects of the Internet, as is the related concept of locality, the idea that local changes should not have a broader impact on the network.

These concepts are critical to the stability and scalability of the Internet. With millions of individual actors engineering different parts of the network, there would be chaos if every change had impact across the entire Internet.

Locality implies that the impact of technical changes made to realize blocking will only be within a defined scope. As discussed above, this scope might be narrow in one dimension (set of users or set of applications affected) but broad in another. Changes made to effectuate blocking are often targeted at a particular locality, but result in blocking outside of the intended scope. For example, web filtering systems in India and China have been shown to cause "collateral damage" by blocking users in Oman and the US from accessing web sites in Germany and Korea [IN-OM-filtering][CCS-GFC-collateral-damage].

3. Examples of Blocking

As noted above, systems to restrict or block Internet communications have evolved alongside the Internet technologies they seek to restrict. Looking back at the history of the Internet, there have been several such systems deployed, with varying degrees of effectiveness.

- Firewalls: Firewalls are a very common form of service blocking, employed at many points in today’s Internet. Typically, firewalls block according to content-neutral rules, e.g., blocking all inbound connections or outbound connections on certain ports. Firewalls can be deployed either on end hosts (under user control), or at network boundaries.

- Web Filtering: HTTP and HTTPS are common targets for blocking and filtering, typically targeted at specific URLs. Some enterprises use HTTP blocking to block non-work-appropriate web sites, and several nations require HTTP and HTTPS filtering by their ISPs in order to block illegal content. HTTPS is a challenge for these systems, because the URL in an HTTPS request is carried inside the secure channel. To block access to content made accessible via HTTPS, filtering systems thus must either block based only on IP address, or else obtain a trust anchor certificate that is trusted by endpoints (and thus act as a man in the middle). These filtering systems often take the form of "portals" or "enterprise proxies." These portals present their own HTTPS certificates that are invalid for any given domain according to normal validation rules, but may still be trusted if the user install a security exception.

- Spam Filtering: Spam filtering is one of the oldest forms of service blocking, in the sense that it denies spammers access to recipients’ mailboxes. Spam filters evaulate messages based on a variety of criteria and information sources to decide whether a given message is spam. For example, DNS Reverse Black Lists use the reverse DNS to flag whether an IP address is a known spam source [RFC5782]. Spam filters are typically either installed on user devices (e.g., in a mail client) or operated by a mail domain on behalf of users.

- Domain name seizure: In recent years, US law enforcement authorities have been issuing legal orders to domain name registries to seize domain names associated with the distribution of counterfeit goods and other allegedly illegal activity [US-ICE]. When domain names are seized, DNS queries for the seized names are typically redirected to resolve to U.S. government IP addresses that host information about the seizure, either by an authoritative server or by an intermediate resolver. The effectiveness of domain seizures is limited by the mobility principle, since the application using the seized name can simply use another name. Seizures can come into conflict with the locality principle, since content is blocked not only within the jurisdiction of the seizure, but globally, even when it may be affirmatively legal elsewhere [RojaDirecta]. When domain redirection is effected via redirections at intermediate resolvers rather than at authoritative servers, it directly contradicts the DNS security architecture [RFC4033].

- Safe Browsing: Modern web browsers provide some measures to prevent users from accessing malicious web sites. For instance, before loading a URL, current versions of Google Chrome and Firefox web browsers use the Google Safe Browsing service to determine whether or not a given URL is safe to load [SafeBrowsing]. The DNS can also be used to mark domains as safe or unsafe [RFC5782].

- Manipulation of routing and addressing data: Governments have recently intervened in the management of IP addressing and routing information in order to maintain control over a specific set of DNS servers. As part of an internationally coordinated response to the DNSChanger malware, a Dutch court ordered the RIPE NCC to freeze the accounts of several resource holders as a means to limit the resource holders’ ability to use certain address blocks [GhostClickRIPE]. These actions have led to concerns that the resource certification system and related secure routing technologies developed by the IETF SIDR working group might be subject to government manipulation as well [RFC6480], potentially for the purpose of denying targeted networks access to the Internet.

4. Blocking Design Patterns

Considering a typical end-to-end Internet communcation, there are three logical points at which blocking mechanisms can be put in place: the middle and either end. Mechanisms based in the middle usually involve an intermediary device in the network that observes Internet traffic and decides which communications to block. At the service end of a communication, authoritative databases (such as the DNS) and servers can be manipulated to deny or alter service delivery. At the user end of a communication, authentication and reputation systems enable user devices (and users) to make decisions about which communications should be blocked.

In this section, we discuss these three "blocking design patterns" and how they align with the Internet architectural principles outlined above. In general, the third pattern -- informing user devices of which services should be blocked -- is the most consistent with the Internet architecture.

4.1. Intermediary-Based Blocking

A common goal for blocking systems is for the system to be able to block communications without the consent or cooperation of either endpoint to the communication. Such systems are thus implemented using intermediary devices in the network, such as firewalls or filtering systems. These systems inspect user traffic as it passes through the network, decide based on the content of a given communication whether it should be blocked, and then block or allow the communication as desired.

Common examples of intermediary-based filtering are firewalls and network-based web-filtering systems. For example, web filtering devices usually inspect HTTP requests to determine the URL being requested, compare that URL to a list of black-listed or white-listed URLs, and allow the request to proceed only if it is permitted by policy (or at least not forbidden). Firewalls perform a similar function for other classes of traffic in addition to HTTP.

It should be noted that these "intermediaries" are often not far from the edge of the network. For example, many enterprise networks operate firewalls that block certain web sites, as do some residential ISPs. In some cases, this filtering is done with the consent or cooperation of the affected users. PCs within an enterprise, for example, might be configured to trust an enterprise proxy, a residential ISP might offer a "safe browsing" service, or mail clients might authorize mail servers on the local network to filter spam on their behalf. These cases are effectively equivalent to the "Endpoint-Based Blocking" scenarios discussed below, since the endpoint has authorized the intermediary to block on its behalf. The challenges discussed in this section arise mostly for scenarios where endpoints are not assumed to cooperate with filtering (i.e., they might have incentives to circumvent filtering).

Accomplishing blocking by using intermediaries conflicts with the end-to-end and transparency principles noted above. The very goal of blocking in this way is to impede transparency for particular content or communications. For this reason, intermediary-based approaches to blocking run into several technical issues that limit their viability in practice. In particular, many issues arise from the fact that an intermediary needs to have access to a sufficient amount of traffic to make its blocking determinations.

The first challenge to obtaining this traffic is simply gaining access to the constituent packets. The Internet is designed to deliver packets from source to destination -- not to any particular point along the way. In practice, inter-network routing is often asymmetric, and for sufficiently complex local networks, intra-network traffic flows can be asymmetric as well.

This asymmetry means that an intermediary will often see only one half of a given communication (if it sees any of it at all), limiting its ability to make decisions based on the content of the communication. For example, a URL-based filter cannot make blocking decisions if it only has access to HTTP responses (not requests). Routing can sometimes be forced to be symmetric within a given network using routing configuration, NAT, or layer-2 mechanisms (e.g., MPLS), but these mechanisms are frequently brittle, complex, and costly -- and often reduce network performance relative to asymmetric routing.

Once an intermediary has access to traffic, it must identify which packets must be filtered. This decision is usually based on some combination of information at the network layer (e.g., IP addresses), transport layer (ports), or application layer (URLs). The communicating endpoints can deny the intermediary access to these by using encryption (see below), but IP addresses must be visible, even if packets are protected with IPsec. However, blocking based on IP addresses is the simplest form of filtering to circumvent, because a filtered site need only change a single DNS record to move all of its services to a new IP address. Indeed, in the face of IP-based blocking in some networks, services such as The Pirate Bay are now using cloud hosting services so that their IP addresses are difficult for intermediaries to predict [BT-TPB][TPB-cloud].

If application content is encrypted with a security protocol such as IPsec or TLS, then the intermediary will require the ability to decrypt the packets to examine application content. Since security protocols are designed to provide end-to-end security (i.e., to prevent intermediaries from examining content), the intermediary would need to masquerade as one of the endpoints, breaking the authentication in the security protocol, reducing the security of the users and services affected, and interfering with legitimate private communication.

If the intermediary is unable to decrypt the security protocol, then its blocking determinations for secure sessions can only be based on unprotected attributes, such as IP addresses and port numbers. Some blocking systems today still attempt to block based on these attributes, for example by blocking TLS traffic to known proxies that could be used to tunnel through the blocking system.

However, as the Telex project recently demonstrated, if an endpoint cooperates with a server, it can create a TLS tunnel that is indistinguishable from legitimate traffic [Telex]. For example, if a banking website operated a Telex server, then a blocking system would be unable to distinguish legitimate encrypted banking traffic from Telex-tunneled traffic to that server (potentially carrying content that the blocking system would have blocked).

Thus, in principle it is impossible to block tunneled traffic through an intermediary device without blocking all secure traffic. (The only limitation in practice is the requirement for special software on the client.) In most cases, blocking all secure traffic is an unacceptable consequence of blocking, since security is often required for services such as online commerce, enterprise VPNs, and management of critical infrastructure. If governments or network operators were to force these services to use insecure protocols so as to effectuate blocking, they would expose their users to the various attacks that the security protocols were put in place to prevent.

Some network operators may assume that only blocking access to resources available via unsecure channels is sufficient for their purposes -- i.e., that the size of the user base that will be willing to use secure tunnels and/or special software to circumvent the blocking is low enough to make blocking via intermediaries worthwhile. Under that assumption, one might decide that there is no need to control secure traffic, and thus that intermediary-based blocking is an attractive option.

However, the longer such blocking systems are in place, the more likely it is that efficient and easy-to-use tunnelling tools will become available. The proliferation of the Tor network, for example, and its increasingly sophisticated blocking-avoidance techniques demonstrate that there is energy behind this trend [Tor]. Thus, intermediary-based blocking becomes less effective over time.

Intermediary-based blocking is a key contributor to the arms race that has led to the development of these kinds of tools, the result of which is to create unecessary layers of complexity in the Internet. Before content-based blocking became common, the next best option for intermediaries was port blocking, the widespread use of which has driven more applications and services to use ports (80 most commonly) that are unlikely to be blocked. In turn, intermediaries shifted to finer-grained content blocking over port 80, content providers shifted to encrypted channels, and intermediaries began seeking to identify those channels. Because the premise of intermediary-based blocking is that endpoints have incentives to circumvent it, this cat-and-mouse game is an invetiable by-product of this form of blocking.

In sum, blocking via intermediaries is only effective in a fairly constrained set of circumstances. First, the routing structure of the network needs to be such that the intermediary has access to any communications it intends to block. Second, the blocking system needs an out-of-band mechanism to mitigate the risk of secure protocols being used to avoid blocking (e.g., human analysts identifying IP addresses of tunnel endpoints), which may be resource-prohibitive, especially if tunnel endpoints begin to change frequently. If the network is sufficiently complex, or the risk of tunneling too high, then intermediary-based blocking is unlikely to be effective, and in any case this type of blocking drives the development of increasingly complex layers of circumvention.

4.2. Server-Based Blocking

[TO DO: Likely to distinguish server-based filtering from infrastructure-based (DNS, RPKI) filtering in a future version.]

Internet services are driven by physical devices such as web servers, DNS servers, certificate authorities, WHOIS databases, and Internet Route Registries. These devices control the structure and availability of Internet applications by providing data elements that are used by application code. For example, changing an A or AAAA record on a DNS server will change the IP address that is bound to a given domain name; applications trying to communicate with the host at that name will then communicate with the host at the new address.

As physical objects, the servers that underlie Internet applications exist within the jurisdiction of governments, and their operators are thus subject to certain local laws. It is thus possible for laws to be structured to facilitate blocking of Internet services operated within a jurisdiction, either via direct government action or by allowing private actors to demand blocking (e.g., through lawsuits).

The "seizure" of domain names discussed above is an example of this type of blocking. Even though some of the affected domain names belonged to non-US entities (e.g., RojaDirecta is Spanish), they were subject to blocking by the US government because certain servers were operated in the US. Government officials required the operators of the parent zones of a target name (e.g., "com" for "example.com") to direct queries for that name to a set of government-operated name servers. Users of services under a target name would thus be unable to locate the servers providing services for that name, denying them the ability to access these services. The action of the Dutch police against the RIPE NCC is of a similar character, limiting the ability of certain ISPs to manage their Internet services by controlling their WHOIS information.

Blocking services by disabling or manipulating servers does respect the end-to-end principle, since the affected server is one end of the blocked communication. However, its interactions with layering, resource mobility, and autonomy can limit its effectiveness and cause undesirable consequences.

The layered architecture of the Internet means that there are several points at which access to a service can be blocked. The service can be denied Internet access (via control of routers), DNS services (DNS servers), or application-layer services (application servers, e.g., web servers). Blocking via these channels, however, can be both amplified and limited by the global nature of the Internet.

On the one hand, the global nature of Internet resources amplifies blocking actions, in the sense that it increases the risk of overblocking -- collateral damage to legitimate use of a resource. A given network or domain name might host both legitimate services and services that governments desire to block. A service hosted under a domain name and operated in a jurisdiction where it is considered undesirable might be considered legitimate in another jurisdiction; a blocking action in the host jurisdiction would deny legitimate services in the other.

On the other hand, the distributed and mobile nature of Internet resources limits the effectiveness of blocking actions. Because an Internet service can be reached from anywhere on the Internet, a service that is blocked in one jurisdiction can often be moved or re-instantiated in another jurisdiction. Likewise, services that rely on blocked resources can often be rapidly re-configured to use non-blocked resources. For example, in a process known as "snowshoe spamming," a spam originator uses addresses in many different networks as sources for spam. This technique is already widely used to spread spam generation across a variety of resources and jursidictions to prevent spam blocking from being effective.

The efficacy of server-based blocking is further limited by the autonomy principle discussed above. If the Internet community realizes that a blocking decision has been made and wishes to counter it, then local networks can "patch" the authoritative data to avoid the blocking. For example, in 2008, Pakistan Telecom attempted to deny access to YouTube within Pakistan by announcing bogus routes for YouTube address space to peers in Pakistan. YouTube was temporarily denied service on a global basis due to a route leak, but service was restored in approximately two hours because network operators around the world re-configured their routers to ignore the blocking routes [RenesysPK]. In the context of SIDR and secure routing, a similar re-configuration could be done if a resource certificate were to be revoked in order to block routing to a given network.

In the DNS context, similar work-arounds are available. If a domain name were blocked by changing authoritative records, network operators can restore service simply by extending TTLs on cached pre-blocking records in recursive resolvers, or by statically configuring resolvers to return un-blocked results for the affected name. Indeed these techniques are commonly used in practice to provide service to domains that have been disrupted, such as the .ht domain during the 2010 earthquake in Haiti [EarthquakeHT]. While the point of these measures was to counter the effects of a natural disaster rather than to counter filtering, the same technical means can also counter the effects of filtering based on modifications to the authoritative server for a domain.

Resources such as the DNS, the RPKI, and the Internet Route Registries are generic technical databases intended to record certain facts about the network. The DNS, for example, stores information about which servers provide services for a given name; the RPKI about which entities have been allocated IP addresses. To offer specialized Internet services and applications, different entities rely on these generic records in different ways. Thus the effects of changes to the databases can be much more difficult to predict than, for example, the effect of shutting down a web server (which fulfills the specific purpose of serving web content).

Server-based blocking also has a variety of other implications that may reduce the stability, accessibility, and usability of the global Internet. Server-side blocking may encourage the development of parallel or "underground" server-side infrastructure, for example. These considerations are further discussed in ISOC’s whitepaper on DNS filtering [ISOCFiltering], but they also apply to other global Internet resources.

In summary, server-based blocking can sometimes be used to immediately block a target service by removing some of the resources it depends on. However, such blocking actions often have harmful side effects due to the global nature of Internet resources. The global mobility of Internet resources, together with the autonomy of the networks that comprise the Internet, can mean that the effects of server-based blocking can be quickly be negated. To adapt a quote by John Gilmore, "The Internet treats blocking as damage and routes around it".

4.3. Endpoint-Based Blocking

Internet users and their devices make thousands of decisions every day as to whether to engage in particular Internet communications. Users decide whether to click on links in suspect email messages; browsers advise users on sites that have suspicious characteristics; spam filters evaluate the validity of senders and messages. If the hardware and software making these decisions can be instructed not to engage in certain communications, then the communications are effectively blocked because they never happen.

There are several systems in place today that advise user systems about which communications they should engage in. As discussed above, several modern browsers consult with "Safe Browsing" services before loading a web site in order to determine whether the site could potentially be harmful. Spam filtering is one of the oldest blocking systems in the Internet; modern blocking systems typically make use of one or more "reputation" or "blacklist" databases in order to make decisions about whether a given message or sender should be blocked. These systems typically have the property that many blocking systems (browsers, MTAs) share a single reputation service.

This approach to blocking is consistent with the Internet architectural principles discussed above, dealing well with the end-to-end principle, layering, mobility, and locality/autonomy.

Much like server-based blocking, endpoint-based blocking is performed at one end of an Internet communication, and thus avoids the problems related to end-to-end security mechanisms that intermediary-based blocking runs into. Endpoint-based blocking also lacks some of the limitations of server-based blocking: While server-based blocking can only see and affect the portion of an application that happens at a given server (e.g., DNS name resolution), endpoint-based blocking has visibility into the entire application, across all layers and transactions. This visibility provides endpoint-based blocking systems with a much richer set of information on which to make blocking decisions.

In particular, endpoint-based blocking deals well with adversary mobility. If a blocked service relocates resources or uses different resources, a server-based blocking approach may not be able to affect the new resources. An intermediary-based blocking system may not even be able to tell whether the new resources are being used, if the blocked service uses secure protocols. By contrast, endpoint-based blocking systems can detect when a blocked service’s resources have changed (because of their full visibility into transactions) and adjust blocking as quickly as new blocking data can be sent out through a reputation system.

Finally, in an endpoint-based blocking system, blocking actions are performed autonomously, by individual endpoints or their delegates. The effects of blocking are thus local in scope, minimizing the effects on other users or other, legitimate services.

The primary challenge to endpoint-based blocking is that it requires the cooperation of endpoints. Where this cooperation is willing, this is a fairly low barrier, requiring only reconfiguration or software update. Where cooperation is unwilling, it can be challenging to enforce cooperation for large numbers of endpoints. If cooperation can be achieved, endpoint-based blocking can be much more effective than other approaches because it is so coherent with the Internet's architectural principles.

5. Summary of Trade-offs

Intermediary-based blocking is a relatively low-cost blocking solution in some cases, but a poor fit with the Internet architecture, especially the end-to-end principle. It thus suffers from several limitations.

- Examples: Firewalls, web filtering systems.
- A single intermediary device can be used to block access by many users to many services.
- Intermediary blocking can be done without the cooperation of either endpoint to a communication (although having that cooperation makes it more likely to be effective).
- Intermediaries often lack sufficient information to make blocking decisions, due to routing asymmetry or encryption.
- Intermediary blocking sometimes involves breaking end-to-end security assurances.
- Tunneling through blocking is difficult to prevent without preventing legitimate secure services.

Server-based blocking can provide rapid effects for resources under the control of the blocking entity, but its ultimate effectiveness is limited by the global, autonomous nature of Internet resources and networks, and it may create undesirable collateral damage to Internet services.

- o Examples: Domain name seizures, WHOIS account freezing, RPKI certificate revocation.

- o Internet services that depend on specific resources can be blocked by disabling those resources.

- o Blocked resources can often be easily relocated or reinstantiated in a location where they will not be blocked.

- o Resources used by undesirable services are often also used by legitimate services, resulting in collateral damage.

- o Autonomy of Internet networks and users allows them to "route around" blocking.

Endpoint-based blocking matches well with the overall design of the Internet.

- o Examples: Safe browsing, spam filtering, enterprise HTTPS proxies (explicitly trusted by clients).

- o Endpoints block services by deciding whether or not to engage in a given communication.

- o Blocking system has full visibility into all layers involved in a communication.

- o Adversary mobility can be quickly observed so that blocking systems can be updated to account for it.

- o Requires cooperation of endpoints.

Because it agrees so well with Internet architectural principles, endpoint-based blocking is the form of Internet service blocking that is least harmful to the Internet. It is likely to be the most effective long-term technical filtering mechanism in many cases.

While this document has focused on technical mechanisms used to filter Internet content, a variety of non-technical mechanisms may also be available depending on the particular context and goals of the public or private entity seeking to restrict access to content. For example, purveyors of illegal online content can be pursued through international cooperation, by using the criminal justice system, and by targeting the funding that supports their activities through collaboration with financial services companies [click-trajectories]. Thus even in cases where endpoint-based filtering is not viewed as a viable means of restricting access to content, entities seeking to filter may find other strategies for achieving their goals that do not involve interfering with the architecture or operation of the Internet.

In reality, the various approaches discussed above are all applied for different reasons. Often, the choice of a filtering solution is constrained by practical limitations on which parts of the network are under the control of the entity implementing filtering, and which parts of the network are trusted to cooperate. For example, an ISP that is subject to filtering requirements might implement an intermediary-based filtering approach because it cannot be sure that endpoints will cooperate in filtering. As discussed above, government agencies tasked with disabling certain foreign web sites have done so by manipulating servers that are within their own jurisdictions, since those are the servers they can access. An enterprise with filtering requirements might require install a certain filtering software package on enterprise-owned PCs.

It is therefore realistic to expect that certain entities will continue to attempt to conduct intermediary- or server-based filtering since they may not have control over the endpoints they wish to affect or because the endpoints do not have incentives to consent to the filtering. In some cases, an approach that combines one of these with endpoint-based filtering can help strike a better balance. For example, a filtering system might make it possible for some endpoints to cooperate or "opt in" to filtering, rather than deploying a purely intermediary-based solution.

Those with a desire to filter should take into account the limitations discussed in this document and wholistically assess the space of technical and non-technical solutions at their disposal and the likely effectiveness of each combination of approaches.

6. Security Considerations

The primary security concern related to Internet service blocking is the effect that it has on the end-to-end security model of many Internet security protocols. When blocking is enforced by an intermediary with respect to a given communication, the blocking system may need to obtain access to confidentiality-protected data to make blocking decisions. Mechanisms for obtaining such access typically require the blocking system to defeat the authentication mechanisms built into security protocols.

For example, some enterprise firewalls will dynamically create TLS certificates under a trust anchor recognized by endpoints subject to blocking. These certificates allow the firewall to authenticate as any website, so that it can act as a man-in-the-middle on TLS connections passing through the firewall.

Modifications such as these obviously make the firewall itself a point of weakness. If an attacker can gain control of the firewall or compromise the key pair used by the firewall to sign certificates, he will have access to the plaintext of all TLS sessions for all users behind that firewall, in a way that is undetectable to users.

When blocking systems are unable to inspect and block secure protocols, it is tempting to simply block those protocols. For example, a web blocking system that is unable to hijack HTTPS connections might simply block any attempted HTTPS connection. However, since Internet security protocols are commonly used for critical services such as online commerce and banking, blocking these protocols would block access to these services as well, or worse, force them to be conducted over insecure protocols.

Security protocols can, of course, also be used a mechanism for blocking services. For example, if a blocking system can insert invalid credentials for one party in an authentication protocol, then the other end will typically terminate the connection based on the authentication failure. However, it is typically much simpler to simply block secure protocols than to exploit those protocols for service blocking.

7. Informative References

[BT-TPB] Meyer, D., "BT blocks The Pirate Bay", June 2012, <http://www.zdnet.com/bt-blocks-the-pirate-bay-4010026434/>.

[CCS-GFC-collateral-damage]
"The Collateral Damage of Internet Censorship by DNS Injection", July 2012, <http://conferences.sigcomm.org/sigcomm/2012/paper/ccr-paper266.pdf>.

[EarthquakeHT]
Raj Upadhaya, G., ".ht: Recovering DNS from the Quake", March 2010, <http://www.apricot.net/apricot2010/__data/assets/pdf_file/0019/19018/Lightning-Talk_03_Gaurab-Upadhaya-dotht-apricot-lightning.pdf>.

[GhostClickRIPE]
RIPE NCC, "RIPE NCC Blocks Registration in RIPE Registry Following Order from Dutch Police", 2012, <http://www.ripe.net/internet-coordination/news/about-ripe-ncc-and-ripe/

ripe-ncc-blocks-registration-in-ripe-registry-following-order-from-dutch-police>.

[IN-OM-filtering] Citizen Lab, "Routing Gone Wild", July 2012, <https://citizenlab.org/2012/07/routing-gone-wild/>.

[ISOCFiltering] Internet Society, "DNS: Finding Solutions to Illegal On-line Activities", 2012, <http://www.internetsociety.org/what-we-do/issues/dns/finding-solutions-illegal-line-activities>.

[RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.

[RFC3724] Kempf, J., Austein, R., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, March 2004.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4084] Klensin, J., "Terminology for Describing Internet Connectivity", BCP 104, RFC 4084, May 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4924] Aboba, B. and E. Davies, "Reflections on Internet Transparency", RFC 4924, July 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5782] Levine, J., "DNS Blacklists and Whitelists", RFC 5782, February 2010.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, February 2012.

[RenesysPK] Brown, M., "Pakistan hijacks YouTube", February 2008, <http://www.renesys.com/blog/2008/02/

pakistan_hijacks_youtube_1.shtml>.

[RojaDirecta]
Masnick, M., "Homeland Security Seizes Spanish Domain Name That Had Already Been Declared Legal", 2011, <http://www.techdirt.com/articles/20110201/10252412910/homeland-security-seizes-spanish-domain-name-that-had-already-been-declared-legal.shtml>.

[SafeBrowsing]
Google, "Safe Browsing API", 2012, <https://developers.google.com/safe-browsing/>.

[TPB-cloud]
"The Pirate Cloud", October 2012, <http://thepiratebay.se/blog/224>.

[Telex] Wustrow, E., Wolchok, S., Goldberg, I., and J. Halderman, "Telex: Anticensorship in the Network Infrastructure", August 2011, <https://telex.cc/>.

[Tor] "Tor Project: Anonymity Online", 2012, <https://www.torproject.org/>.

[US-ICE] U.S. Immigration and Customs Enforcement, "Operation in Our Sites", 2011, <http://www.ice.gov/doclib/news/library/factsheets/pdf/operation-in-our-sites.pdf>.

[click-trajectories]
Levchenko, K., Pitsillidis, A., Chacra, N., Enright, B., Felegyhazi, M., Grier, C., Halvorson, T., Kreibich, C., Liu, H., McCoy, D., Weaver, N., Paxson, V., Voelker, G., and S. Savage, "Click Trajectories: End-to-End Analysis of the Spam Value Chain", 2011, <http://cseweb.ucsd.edu/˜savage/papers/Oakland11.pdf>.

Authors' Addresses

Richard Barnes
BBN Technologies
1300 N. 17th St
Arlington, VA 22209
USA

Phone: +1 703 284 1340
Email: rbarnes@bbn.com

Alissa Cooper
CDT
1634 Eye St. NW, Suite 1100
Washington, DC 20006
USA

Email: acooper@cdt.org

Olaf Kolkman
NLnet Labs

Email: olaf@nlnetlabs.nl