The (R)evolution of the Internet Protocol Suite

Brian K. Haberman

ABSTRACT

The allegations of widespread, pervasive monitoring of Internet traffic have ignited new thrusts to strengthen the core protocols that underpin the Internet. Many of these protocols were developed in a time when the community of interest was small and trust between the members was rock solid. Such trust led to protocols that did not protect against actions such as denial-of-service attacks, man-in-the-middle attacks, or pervasive monitoring, leading to the widespread use of relatively insecure protocols. The Internet is no longer a homogenous, all-trusting community, and the underlying protocols are being examined to determine whether they should undergo potentially drastic changes to address the threats. This article highlights some of the key changes being considered and describes their possible benefits and drawbacks. The impact of these changes could be beneficial to Internet users concerned about indiscriminate eavesdropping but could also prove costly to innocent network operators/users who have developed operational and use models based on the older, less secure designs of the protocols.

INTRODUCTION

The origins of the Internet arose from a classically academic research project. Despite the ever-growing size of the Internet research community, the protocols developed all retained a core aspect from the beginning—implicit trust. The Internet was able to grow because independent teams of researchers and developers trusted the information being shared across their common medium. Collaboration, and the trust it grew, sped innovation at rates faster than could have been imagined. A classic example of this collaborative environment is the Robustness Principle attributed to Jon Postel:[1] "Be liberal in what you accept, and conservative in what you send." The Robustness Principle epitomizes the belief that all involved are working toward a common goal.

The implied trust and belief in collaboration works well when all parties involved have a shared goal. In the case of the Internet protocol suite, that shared goal has been the development of a scalable and functional global network. However, the parameters of that network have changed as it has transitioned from research project to commercial venture to indispensable infrastructure. Not all actors share the same goal. Not all goals have positive outcomes for all involved. However, many of the key underlying Internet protocols have not evolved to address the emergence of these competing goals. Many protocols have either limited or no protections against a variety of attacks. Many operators are wary of enabling security on some protocols because of a fear of interoperability issues or an unwillingness to increase costs without some return on investment. There are a myriad of reasons for the general lack of security in key protocols that underpin the Internet.

With the emergence of allegations of widespread, pervasive monitoring, segments of the Internet population have reacted swiftly to address the shortcomings in the protocol suite. Bruce Schneier, a noted security researcher, challenged the Internet Engineering Task Force (IETF), the body charged with maintaining the Internet protocol specifications, to strengthen the core protocols against "wholesale spying."[2] In response, the IETF developed a "statement of principles" in 2014 designed to guide protocol development going forward.[3] Two items within that document are worth noting specifically:

- The IETF defines pervasive monitoring as a widespread attack on privacy.
- The IETF will work to mitigate pervasive monitoring.

The IETF began work to mitigate pervasive monitoring almost immediately after Edward Snowden's initial allegations of pervasive monitoring,[4] before members completely agreed on the statement of principles document. Subsets of the community immediately began considering potential changes in key protocols in response to the allegations.

The remainder of this article describes three representative efforts to minimize the impact of pervasive monitoring on Internet users. These efforts were selected because of their direct impact on everyday users. As a part of the discussion, this article discusses the key benefits and key drawbacks of the proposed approaches. These trade-offs highlight the tricky nature of retrofitting security into existing, deployed, and widely used network protocols. It should be noted that these three cases are representative of a broader effort within the IETF to strengthen new and existing network protocols. Other protocols such as Internet Protocol version 6 (IPv6), e-mail, Network Time Protocol (NTP), Border Gateway Protocol (BGP), and Transport Layer Security (TLS) are all being studied or revised to strengthen their security posture.

Hypertext Transfer Protocol 2.0

In 2015, a large majority of Internet web traffic is carried via version 1.1 of the Hypertext Transfer Protocol (HTTP),[5] which was standardized in 1999. The IETF is currently working on a replacement for HTTP 1.1, designated HTTP 2.0. As a part of the discussion of the structure and semantics of HTTP 2.0, consideration is being given to encrypting *all* web traffic. In today's Internet, only sensitive web traffic (e.g., online banking) is encrypted. Users recognize this by the use of Uniform Resource Locators (URLs) that start with HTTPS://, as opposed to the unprotected websites that are accessed via HTTP://. The proponents of such a change argue that encryption is one way to mitigate pervasive monitoring of Internet web traffic. Several key arguments can be made that justify having all web traffic transit encrypted connections between web servers and clients (Fig. 1).

The primary argument for encrypting all HTTP 2.0 traffic is that users now expect their data to be protected in the face of pervasive monitoring. The mounting allegations of widespread data collection by a variety of gov-

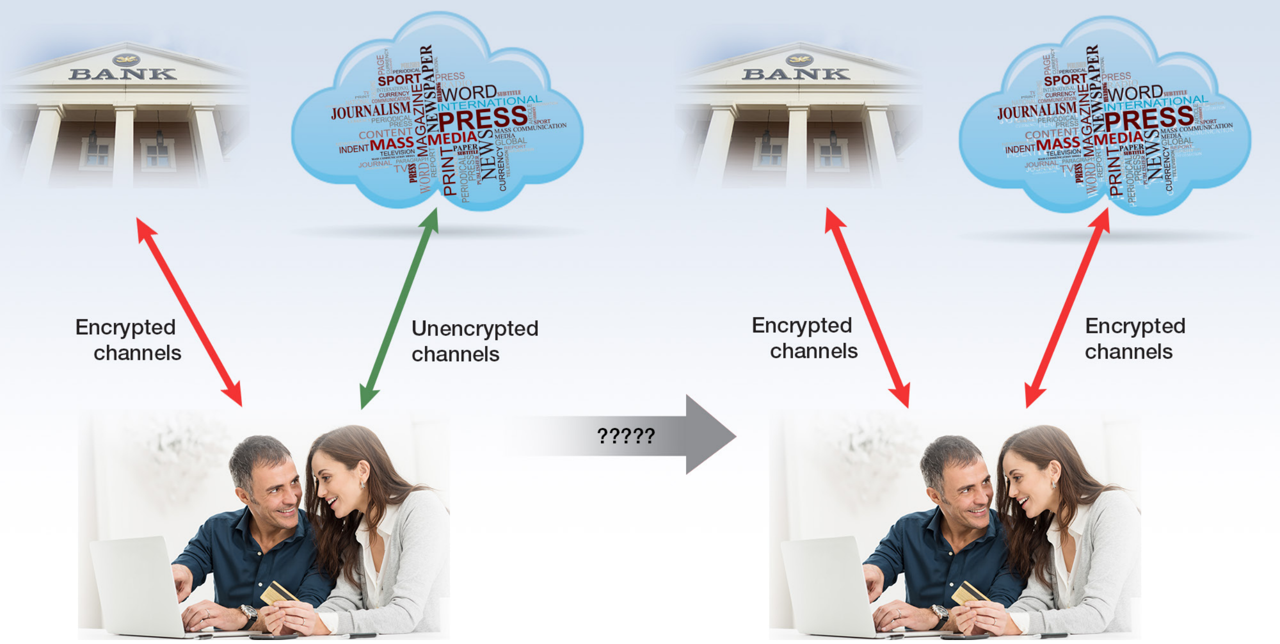

Figure 1. Potential change to web encryption during the transition from HTTP 1.1 to HTTP 2.0.

ernments have sensitized users to the potential loss of privacy. From an interoperability perspective, it is more feasible to standardize the encrypted data exchanges within the protocol than it is to expect ad hoc mechanisms to be developed and widely used. By incorporating encryption into all web exchanges, the HTTP 2.0 standard would simplify the operation of the protocol and protect the privacy of all data.

The next argument is one of economics. Currently, pervasive monitoring is cheap and easy. Bruce Schneier's challenge to the IETF included a goal to "make surveillance expensive again."[2] If HTTP 2.0 encrypts all traffic, as the percentage of web traffic carried by HTTP 2.0 increases, the less practical widespread surveillance becomes. There is an economic reality on the users' side as well. It is generally agreed that the user interface for web security is hard for most users. The use of HTTPS:// URLs does not necessarily mean that all traffic is encrypted. By moving to a privacy-always model, users do not risk their financial future to a clunky user interface. A single model of operation is much simpler to develop, maintain, and operate.

A third argument in favor of encrypting all web traffic is the nebulous definition of privacy.[6] Privacy has become an important concept, but experts cannot agree on a definition of privacy or on which information needs to be kept private. This debate makes choosing what should be private and what should not be private context dependent. However, the *transport* of information should not be tied to context. The application delivering the information should always err on the side of protecting information. Accepting these two tenets removes the need for any hard choices on the part of the user or the content provider when sending data over a web connection.

As with most technology decisions, there are trade-offs and differing opinions. Although the above-described advantages appear straightforward, an encrypted-always web comes with incurred costs, potential drawbacks, and dissenting opinions.

The biggest drawback raised with an always-encrypted web is the impact it would have on existing infrastructure. Most notably, many Internet service providers rely on the use of web caches to reduce the amount of bandwidth needed to serve popular web content. These devices inspect HTTP exchanges and cache the returned content for use by other local users. Caching can dramatically reduce the amount of network traffic generated and speed up response time for popular content. If all web exchanges were encrypted, these caching devices would be unable to operate, increasing the amount of network traffic and increasing the response time. Services such as load balancers, malware scanners, and content filters would also be adversely impacted by a move to an encrypted web. Additionally, policy enforcement devices used to limit web accessibility for legal or business reasons, which require a man-in-the-middle capability, would require a new operational paradigm.

A subset of the community believes it is inappropriate to encrypt all data given that most data are not sensitive (e.g., sports scores). It could be considered excessive to establish the necessary security association between the web server and the client to exchange data that are clearly meant to be public. Additionally, the existing TLS protocol is not designed to handle newer forms of web content such as video and other multimedia, which could make encrypting all web traffic impossible in the near term.

A third counter to encrypting all web traffic is the issue of cost. Encryption technology is complex and requires additional resources (bandwidth, processing, etc.). Current web encryption relies on a certificate-based infrastructure that some see as brittle and already corrupted by entities performing pervasive monitoring. On the evolving side, many view the use of TLS as infeasible for the Internet of Things. Expecting a light switch to have the processing capability to instantiate and maintain security associations seems excessive to some.

At the time of this writing, the direction of the HTTP 2.0 specification has not been decided. The above-described (and other) trade-offs have elicited thoughtful discussion of technical, ethical, legal, and political issues that will impact the standardization effort. Clearly, the people involved in HTTP 2.0 are carefully considering all of the issues related to pervasive monitoring and recognize the impact their decisions will have on the Internet.

DOMAIN NAME SYSTEM PRIVACY

The Domain Name System (DNS)[7] may be one of the most important pieces of Internet infrastructure that most people take for granted, ignore, or do not fully understand. Almost all Internet activities begin with a DNS query to map a target destination's name (e.g., www.ietf.org) to its Internet Protocol (IP) address, allowing the client's network stack to correctly address the connection request. The DNS is a hierarchical distributed database that maps domain names with a variety of information (including IP addresses). The maintenance of the mapping records is delegated to the authoritative name servers designated for each domain name. Although there are several modes of operation, the following is a representative model of DNS exchanges:

1. A client (stub resolver) issues a query for www.example.com to its local recursive resolver.
2. The recursive resolver performs a series of DNS queries, starting at the root of the DNS hierarchy, until it receives an answer from the authoritative name server.
3. The recursive resolver caches the answer for future use and sends the answer back to the client.

As with other Internet protocols, DNS was developed with little security in mind. In fact, the information contained within the DNS should be considered public. If it were not, the Internet would not function as well as it does because the name-to-address mapping function underpins the entire Internet. How many users would want to memorize the IP addresses of their favorite websites? Treating all questions as equal has allowed DNS to remain a relatively simple query/response protocol. But, as mentioned, the rules have changed because of the differing goals of the various Internet participants, and protection of DNS data is being examined intently.

The public nature of DNS data allows network users to efficiently reach the content of interest without knowing anything about the topology of the Internet. However, there are still privacy aspects to be considered. The DNS does not provide users with a search capability. That means that the stub resolver has to know what to ask for to receive a useful answer. So, although the mapping of www.example.com to 2001:500:8d::53 is public information, the fact that a particular user requested that mapping may be sensitive and should be kept private. Additionally, the DNS is being used to map more than just IP addresses to names. For example, DNS records identify: (*i*) the mail server for a particular domain, (*ii*) DNS name servers for a target domain, and (*iii*) security certificates for a target domain name or IP address. Accessing those additional mappings may reveal sensitive metadata about the user formulating the DNS question(s).

The DNS protocol does not currently provide users with any type of privacy protection for the questions asked or the answers received. All DNS exchanges are sent unencrypted and are visible to any entity capable of examining packets in transit. That limitation is true regardless of whether the exchange is between a stub resolver and a recursive resolver or between a recursive resolver and authoritative name servers. The DNS Security Extensions (DNSSEC)[8] only provide for information integrity and authentication, not confidentiality for information exchanges (see Fig. 2).

At the time of this writing, the IETF is actively discussing addressing the lack of privacy controls within the DNS protocol. These discussions are focused on the privacy risks for the user initiating the DNS requests in the face of pervasive monitoring. As noted above, DNS questions may be sensitive based on who is asking for a particular DNS mapping. A passive observer can extract from a DNS request going from the stub resolver to the local recursive resolver:

1. The IP address of the requesting machine (which may map to a specific user)
2. The exact DNS name being queried
3. The type of record being requested (e.g., a mail server record)

The current DNS model levies differing requirements for privacy based on the point where pervasive monitoring may be incurred. Traffic from the stub resolver to a local recursive resolver will contain the exact DNS name being queried as well as the IP address of the specific machine making the request. However, if that query is sent from the local resolver to the authoritative name server, the requesting IP address is that of the local resolver rather than of the original requester. That type of separation may lend itself to different types of privacy controls depending on what type of DNS resolver is originating the DNS request. But that separation does not always hold because it is quite easy for users to operate their own local recursive resolvers, rather than a stub resolver, on their own machines.

The integral nature of DNS on the functioning of the Internet makes the problem of confidentiality a difficult one. DNSSEC eventually gained traction because the additional certificate information was incorporated as yet another DNS record type, and resolvers that did not support DNSSEC could skip the validation. In other words, the functionality of DNSSEC did not affect legacy devices. It is unclear how confidentiality can be incorporated without impacting the operational model of DNS. Many of the pros and cons listed for HTTP 2.0 also apply to DNS confidentiality.

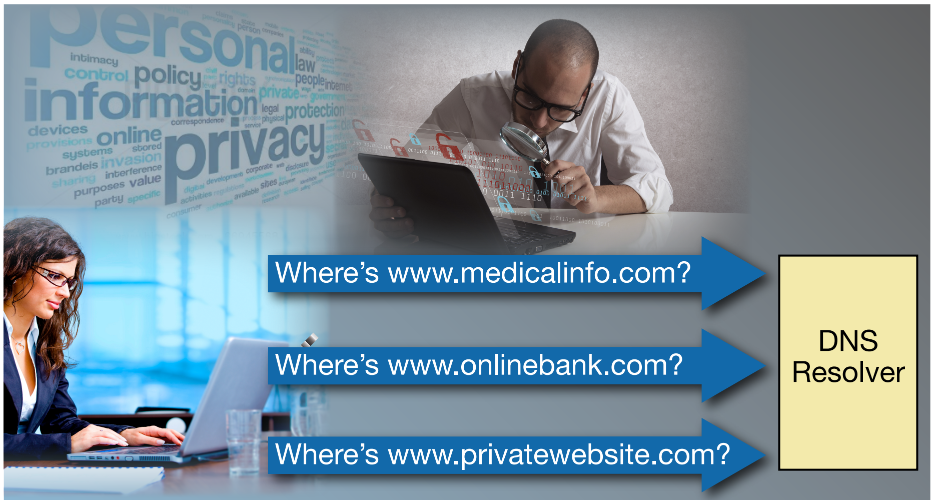
personal
information
privacy
policy
control
online
Where's www.medicalinfo.com?
Where's www.onlinebank.com?
Where's www.privatewebsite.com?
DNS Resolver

Figure 2. Intermediaries snooping on DNS exchanges build substantial profiles of users.

- Not all requester's IP address to queried domain name mappings are sensitive.
- Fundamental changes to the protocol will impact legacy devices.
- Additional infrastructure for confidentiality approaches (e.g., encryption) incurs costs.

These issues and the potential privacy gains from augmenting DNS will drive the discussion within the IETF. The threat to privacy from pervasive monitoring of the DNS is still not completely understood. In some instances, pervasive monitoring provides protection from threats such as malware.[9] However, the decisions made at the DNS protocol level need to consider all aspects and threats to privacy.

USING TLS IN APPLICATIONS

Many application protocols have defined methods for using TLS to encrypt traffic and authenticate one or both endpoints in a communications session. However, there is significant diversity in the definitions of those methods as well as variations in the requirements for the use of TLS. This diversity has led to confusion within the implementation community, resulting in a lack of interoperability and deployment of TLS-protected applications.[10]

The Using TLS in Applications (UTA) working group within the IETF was chartered to address this confusion and simplify the lives of developers wanting to use TLS in application protocols. If standardized methods of use are available, it will be easier for application programmers to develop interoperable implementations of TLS-enabled services. As a starting point, the UTA working group has four preliminary work items to facilitate the use of TLS within applications protocols:

1. Update the definitions for using TLS over a set of representative application protocols. These definitions include communication with proxies, between servers, and between peers, where appropriate, in addition to client/server communication.
2. Specify a set of best practices for TLS clients and servers, including but not limited to recommended versions of TLS, using forward secrecy, and one or more cipher suites and extensions that are mandatory to implement.
3. Consider, and possibly define, a standard way for an application client and server to use unauthenticated encryption through TLS when server and/or client authentication cannot be achieved.
4. Create a document that helps application protocol developers use TLS in future application definitions.

The above-described work items should provide application developers strategic guidance for using TLS to protect critical applications that are susceptible to pervasive monitoring. As a starting point, the e-mail (SMTP, IMAP, and POP), instant messaging (XMPP), and web (HTTP 1.1) protocols will be used as the representative application protocols mentioned in point no. 1 above.

Around 1994, the IETF started requiring all proposed standards to contain a *Security Considerations* section. The premise of such a section was to facilitate the discussion of security issues related to the protocol being specified. This discussion routinely mentions means to mitigate the identified security issues related to the protocol. Unfortunately, many documents simply stated that IP security (IPsec) should be used to mitigate these vulnerabilities. The problem with such generic advice is that it was not useful. In some cases, a description of *how* to use IPsec to mitigate a vulnerability was missing. In other cases, specifications assumed IPsec could protect against anything, when in reality, it could not. In time, the phrase "just use IPsec" became a punch line rather than a solution. To avoid a similar situation with TLS, the UTA working group is formulating a set of recommendations for the proper use of TLS and the datagram version of TLS (called DTLS). These recommendations currently discuss issues related to versions of the protocols, cipher suites to use with the protocols, capability negotiation, and public key lengths. It is envisioned that such guidance will lead to more robust implementations of TLS-protected applications and a wider use of TLS to protect against pervasive monitoring.

Unlike the previously described examples, the outcome of the UTA effort appears much clearer. Many application developers already have some semblance of TLS support within their code bases. This clarity and familiarity makes the outcome of the UTA working group more useful in the short term and more likely to be adopted quickly by both the development community as well as the user community.

DISCUSSION

Whether one calls these protocol changes evolutionary or revolutionary, one thing is clear. There is a potential sea change over the horizon. While some have argued that pervasive monitoring has been occurring for a long time, these recent allegations have become a catalyst for change within the Internet community.[3] The currently proposed changes are only representative of the potential changes being considered for the Internet protocol suite. The IETF's statement of principles[3] on the topic of pervasive monitoring sums up the new model of network protocol standardization: "The IETF will work to mitigate the technical aspects of PM [pervasive monitoring], just as we do for protocol vulnerabilities in general."

The key word in the above statement is *technical*. The Internet community recognizes that there are multiple facets to the pervasive monitoring issue (technical, political, social, legal, etc.) but realizes the technical issues can be worked strictly from an engineering perspective. Even within the purview of network protocol standardization, there are numerous avenues of interest for technical contributions.

As noted in the earlier examples, significant work is ongoing to re-engineer protocols to strengthen them against pervasive monitoring. Historically, far too little effort has gone into the security aspects of network protocols. That mindset appears to be changing. With such re-engineering, additional work will need to be undertaken to address both the interoperability issue with legacy systems and the potential impact on the operational paradigms that were developed while using older, less secure versions of the protocols.

Part of the weakness in current network protocols has been the lack of attention given to protecting private information. That lack of attention is generally caused by two things: first, protocol engineers focusing only on the on-the-wire protocol operation and not considering the value of the information being shared; and, second, the complexity of what privacy entails. The former can be rectified as a part of the standardization process by including privacy-related issues in the review process. In fact, that change has begun as more reviewers provide feedback on privacy issues with protocol proposals being put forth for publication. The latter issue is one of education, and that has begun as well with the publication of a privacy considerations document for Internet standards.[11] As more protocol engineers understand privacy issues and consciously consider them in their design decisions, newer network protocols will be less likely to leak private information as a part of normal protocol operation.

Many people recognize that openness can create an environment of trust. To that end, there has been an increased interest in ensuring that as much of the network infrastructure is open as possible. Although most readers will recognize terms such as *open source* and *open standards*, these terms do not cover the entire spectrum. Open protocol standards allow implementers, researchers, and users to review the protocol operation as required to ensure correct protocol behavior. Open-source software allows people to examine the code to ensure correct operation and behavior. Openness applies to security as well. Transparency in all aspects of the security model or mechanism provides accountability. That accountability allows users to exercise checks and balances that can determine the effectiveness of the security mechanism and expose any potential abuse of the security mechanism. A paradigm of openness will require changes in the behaviors of both people and institutions. Currently, that paradigm shift appears to be gaining momentum within the networking community.

Although these dramatic changes are unfolding, they are not without cost. As mentioned earlier, some of these changes are being resisted primarily because of the impact they will have on current industries, network operators, and users. Several industries have blossomed in the current network environment where most network traffic is free for the analysis. Marketing companies analyze network traffic so that they can provide targeted ads. Malware and spam detection devices perform deep packet inspection of transit traffic within networks. Some Internet service providers redirect users to advertisement-oriented web servers when they mistype domain names. These business practices are predicated on being able to inspect network traffic. Changes to network protocols to enhance privacy, generally through some type of encryption, have the potential to dramatically impact such business models.

Network operators leverage a number of pervasive monitoring techniques in the name of efficient network management. A variety of protocols and tools are widely used that assess bandwidth consumption, traffic profiles, and trends. Any significant increase in the amount of encrypted traffic could affect network operations in ways such as:

1. Increasing bandwidth consumption from both security association signaling and per-packet overhead
2. Increasing packet processing costs for encrypted control and management traffic; and
3. Increasing operational costs to overhaul network management models.

Outside of these costs, network operators will be on the front lines of protocol interoperability issues. Transitioning protocols is recognized as a difficult task. Users and network services are not all managed by one entity, so legacy interoperability will be a critical component of making enhanced protocols viable.

One of the most significant concerns mentioned in relation to increased privacy protections within network protocols is the impact on the everyday user. Most Internet users do not have an understanding of the existing security models in use today. The appearance of a lock icon on their browser makes many people feel they are secure. It is unclear how people will react if their HTTP 2.0-capable browser indicates that *all* sessions are "locked" or the browser does not make any indications at all. Will users recognize that they need to be involved in determining when certain information should be considered private? There will be a large human factors aspect to any dramatic shift in the network protocols if those changes expose new decisions to users.

Despite these costs, the network protocol environment appears poised for change. The lack of a common goal among all stakeholders will preclude the standardization of network protocols without security as an integrated component. That change will not come overnight, but the consensus is that such a change is needed. The impact of the change is still an unknown as the full spectrum of the change is still evolving. Because pervasive monitoring is viewed as more than just a technical problem, other changes from outside the technical community can affect the technical aspects being addressed within the standards community. From all appearances, the evolution of the Internet protocol suite has begun. Its effect on the current Internet is to be determined.

REFERENCES

[1] Braden, R., *Requirements for Internet Hosts—Communication Layers*, Internet Engineering Task Force (1989).

[2] Schneier, B., "Take Back the Internet," *Schneier on Security* (blog), https://www.schneier.com/blog/archives/2013/09/take_back_the_i.html (15 Sep 2013).

[3] Farrell, S., and Tschofenig, H., *Pervasive Monitoring Is an Attack*, Internet Engineering Task Force (2014).

[4] Ball, J., "Edward Snowden NSA Files: Secret Surveillance and Our Revelations So Far," *The Guardian*, http://www.theguardian.com/world/2013/aug/21/edward-snowden-nsa-files-revelations (21 Aug 2013).

[5] Fielding, R., Gettys, J., Mogul, J., Nielson, H. F., Masinter, L., et al., *Hypertext Transfer Protocol—HTTP/1.1*, Internet Engineering Task Force (1999).

[6] Solove, D. J., *Understanding Privacy*, Harvard University Press, Cambridge, MA (2008).

[7] Mockapetris, P., *Domain Names—Implementation and Specification*, Internet Engineering Task Force (1987).

[8] Arends, R., Austein, R., Larson, M., Massey, D., and Rose, S., *Protocol Modifications for the DNS Security Extensions*, Internet Engineering Task Force (2005).

[9] Zhu, Z., Lu, G., Chen, Y., Fu, Z., Roberts, P., and Han, K., "Botnet Research Survey," in *Proc. 32nd Annual IEEE International Computer Software and Applications Conf.*, Turku, Finland, pp. 967–972 (2008).

[10] Using TLS in Applications (UTA) Working Group, *Charter for Working Group*, http://datatracker.ietf.org/wg/uta/charter/ (12 Nov 2013).

[11] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., et al., *Privacy Considerations for Internet Protocols*, Internet Engineering Task Force (2013).

THE AUTHOR

Brian K. Haberman is a member of the Principal Professional Staff and a research scientist in the Tactical Wireless Systems Group. His research interests include IPv6, IP multicast, ad hoc and sensor networks, routing, routing protocol security, and network architecture. He currently serves as an area director within the Internet Engineering Task Force and is a member of the ACM. Prior to working at APL, he held network protocol development, platform design, system architecture, and research positions for several networking companies. His e-mail address is brian.haberman@jhuapl.edu.