The Department of Commerce Internet Policy Task Force (IPTF) issued a formal Request for Comments (RFC) in the Federal Register.[1] This issuance requested feedback on a number of computer security topics related to networking, cybersecurity, and data privacy. This document includes responses to the following items:

- Security Challenges
- Network and Infrastructure Security
  - Malware Mitigation
- Web Security and Consumer Trust
  - Web Security
  - Trusted Downloads
  - Trusted Uploads
  - Cybersecurity and the Internet of Things
  - Internet of Things and Bloatware
- Business Processes and Enabling Markets
  - Vulnerability Disclosure
  - Promoting Participation
  - Potential Workshops or Events

Security Challenges

There are a number of key issues related to information disclosure and corporate responsibility that must be addressed in order to develop a successful security solution. Individual companies and market sectors are not isolated. In today's online world, they are intertwined and codependent in ways that many of them do not recognize. For example:

- *Information disclosure*. Applications typically leak a large amount of personal information. The developer of a protocol or application may not see any risk in these information leaks, but the leaks can dramatically impact the security of other sites and services. With a simple network connection to a web site, an application can disclose geolocation information, system information, and even details about the client's computer platform and whether it is infected with malware.

  This initial type of information leakage is passive. The simple act of connecting to a server discloses information about the user. The network address reveals geolocation hints and application headers disclose operating system and version information.

  With indirect network queries, services can augment the information. For example, the DNS, WHOIS, and ASN databases provide details about the service providers that

[1] https://www.federalregister.gov/articles/2015/03/19/2015-06344/stakeholder-engagement-on-cybersecurity-in-the-digital-ecosystem

clients use. Social networks and search engines can gather your interests and social relationships. HTTP “referer” (sic) headers divulge your preferences to third-party advertisers and data analytic services. Additional active scanning of the client’s system and client-side JavaScript can gather even more personal information.

We cannot assume that users know what their applications and network connections are revealing. They may think that turning off GPS on their smartphone is good enough to hide their location, or that using anonymous services protect their identities, but they are wrong. In many cases, turning off tracking functionality does not actually turn off the functionality. At noted by CNet’s Sharon Profis in 2014, “even when Wi-Fi is disabled, a phone could still be searching for networks.”[2] Moreover, there are no laws or regulations that say companies and services cannot collect this information. The corporate mentality is typically “You gave it to me, so I'm going to collect and use it.”

- *Personal information*. Consumers often volunteer personal information when asked. They will readily provide this data if they are told that it is "for security purposes". For example:

  > "For security, what is your date of birth?"
  > "For security, what is your mother's maiden name?"
  > "For security, what is your social security number?"

  Conventional wisdom is that this vulnerability can be overcome by "educating users". However, this is a falsehood. If educating users worked, then this problem would have been resolved long ago.

  If you ask any user if they should email their social security number to someone they have never met, they will immediately say "no"... and then they will do it anyway. The reason these scams work is a perspective issue called *inattentional blindness*.[3] It is easy to see something happening to other people, but it is hard to see it happening to yourself. For example, you may clearly see that your friend is in a bad relationship, but you cannot see that you are in the same kind of bad relationship. Or, you can clearly see that your friend is falling for an online scam, but you cannot see that you are falling for the same kind of scam.

  If users should not be permitted to do an insecure action, then there should be technical methods deployed that prevent the user from performing an insecure action. This is one of the basic tenants of secure programming: validate inputs.[4]

[2] http://www.cnet.com/how-to/how-to-get-better-battery-life-on-android/
[3] http://en.wikipedia.org/wiki/Inattentional_blindness
[4] https://www.owasp.org/index.php/Data_Validation#Data_Validation_Strategies

- *Information value*. Consumers have no idea what is valuable. We, in the computer security field, often hear users say that they do not care if someone compromises their computer because it contains is nothing of value.[5] These users do not realize that simply having an accessible computer is valuable. Even something as basic as providing your name or gender has value to someone else.

  People want to trust. But in today's online world, information has value. Regardless of whether it is intentional or unintentional, users and their applications are just giving away valuable personal information. Then, when there is a significant data compromise, users complain about the amount of data that was collected.

  Before we can address the issue of data theft, we must address the issue of data value. There are at least three possible options:
  1. Remove the value associated with the data,
  2. Increase the penalty for retaining data such that the data becomes too risky to retain, or
  3. Develop technologies and standards that prevent personal information disclosure and leakage.

  Currently there exists tools and technologies to identify common programming errors, memory leaks, and data validation issues. However, there are no technologies for identifying personal information leakage or to evaluate whether a service needs to know explicit details about the consumer. For example: does every web browser really need to disclose the browser version and operating system details to every web site? This is a concern since the version information directly identifies the system’s patch level and possible vulnerabilities.

- *Incentive to protect*. Software providers and online services do not value user privacy and only value personal information for its resale value. These companies see how to monetize user information. So in this regard, they know the value of information.[6]

  However, these same providers and services make virtually no attempt to protect customer information beyond protecting the resale value. And since there are no regulations to protect the information, these companies give minimal effort toward protecting the information.

[5] Examples include https://wiki.albany.edu/display/public/askit/Securing+the+Human-UAlbany, https://www.cu.edu/sites/default/files/attachments/Module01-YouAreTheTarget-Newsletter.pdf, and http://www.sfasu.edu/itsecurity/docs/Information_Security_Shorts.pdf.

[6] http://www.huffingtonpost.com/2012/05/17/fbme-shows-how-much-youre-worth-to-facebook_n_1523884.html, http://mashable.com/2012/05/14/val-you-calculator-worth-facebook/, and http://thenextweb.com/insider/2013/09/17/whats-the-true-value-of-your-personal-data-meet-the-people-who-want-to-help-you-sell-it/

- *How to protect*. While online services collect a significant amount of personal information, they frequently cannot decide what to protect or how to protect this data.

  For example, many ecommerce sites strive to maintain PCI-DSS compliance. However, it is widely known that the Payment Card Industry's Data Security Standard is nothing more than the absolute minimum, and it is not enough to protect consumer information.[7] Outside of ecommerce, the United States currently has no standards or guidelines for protecting generic consumer data, such as names, email addresses, and other personal information.

- *Reused information*. Service providers can tell users to not use the same password anywhere else, but we all know that users reuse passwords. By the same means, you can use your mother's maiden name at every site that asks for it as a security question. Meanwhile, services continue to simplify the process of linking accounts. For example, Facebook has options to link to Twitter, Google, and accounts at other online services.

  This reuse of information creates a chaining effect between linked services. A simple compromise at one site can provide information needed to compromise another site. As an example, a reporter a *Wired* documented how a minor information leak at Amazon allowed an attacker to compromise his Apple account, and the Apple account permitted compromising Google and Twitter accounts.[8]

  Many companies do not appear to realize how intertwined they really are. Facebook thinks they are secure because they use a phone for secondary validation. With caller-id spoofing,[9] an attacker can take over a cell phone account and then reset the Facebook password. Most online users link everything to everything. A compromise of a cell phone will compromise a user's Facebook account and the link can be exploited to compromise Twitter, Google, Flickr, and other accounts.

A lasting resolution to these key issues is unlikely to come about by addressing siloed topics. Mitigating botnets, providing more secure download options, strengthening security for the Internet of Things, etc. are all topics requested by the Department of Commerce in this RFC. However, these approaches address symptoms and are temporary solutions at best. This interdependency of services and protocols, leaking of personal information, and chaining of services will undermine any temporary solutions.

While the Department of Commerce is not a regulatory organization, they do have the ability to facilitate public discussion between disparate groups. Having corporations, service providers, and software engineers develop and adopt strong standards for compliance is a

[7] http://www.verizonenterprise.com/resources/report/rp_pci-report-2015_en_xg.pdf
[8] http://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/
[9] http://en.wikipedia.org/wiki/Caller_ID_spoofing

start. However, current regulations detract from the ability to adopt widespread solutions. In order to better secure the general public's online information we need to:

1. *Remove software patents*. Software patents restrict the ability to deploy and adopt solutions. In particular, some patents prevent deploying viable security-oriented technologies.[10]

2. *Break up telecommunication and networking monopolies*. Right now, consumers have multiple options for social networks and many options for email providers. However, as noted by FCC Chairman Tom Wheeler in 2014,[11] most consumers in the United States have only one option for broadband Internet access; the broadband providers have effectively divided up the United States into regional monopolies. Due to chaining, online servers are only as secure as the local networks. If consumers had more options for local service providers, then consumers could choose a provider that offers the best security and the least amount of information leakage.

3. *Steep fines for compromises*. As noted in the chaining example,[12] an information leakage at one company can lead to compromises at other companies. What happens if someone uses information from Amazon to compromise an Apple account? Apple may tell users to change passwords[13] or issue a simple *mea culpa*[14]. Amazon closed their vulnerability,[15] while Google and Twitter effectively blamed the user for having compromised information (even though both companies provide options to help users link accounts).

   Each of these companies should share any penalties. When the penalties outweigh the value to their corporate reputations and the value of the information, these companies will begin to do everything possible to protect their infrastructure. But until we reach that financial tradeoff, where developing a secure infrastructure costs less than the penalty for a compromise, these companies have no incentive to protect consumers from information leaks, no incentive to collect less personal information, and no incentive to strengthen their security profile.

[10] Examples of patents around security solutions include http://www.eweek.com/security/ibm-patents-cloud-app-security-solution-for-mobile-devices.html, http://www.spyrus.com/new-spyrus-patents-transparently-upgrade-legacy-security-solutions-with-advanced-cryptographic-algorithms/, and http://www.waterfall-security.com/waterfall-security-granted-patent-for-secure-implementation-of-network-based-sensors/. We do not know if any of the security solutions mentioned in these examples work. But if they work, then the patents prevent widespread adoption.
[11] https://apps.fcc.gov/edocs_public/attachmatch/DOC-329160A1.pdf
[12] http://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/
[13] http://www.cnet.com/news/apple-urges-hacked-users-to-change-passwords/
[14] http://www.cbsnews.com/news/apple-amazon-make-changes-after-journalists-hack/
[15] http://www.wired.com/2012/08/apple-icloud-password-freeze/

4. *Provide immunity for vulnerability reporting*. Far too often we hear about security risks being disclosed to companies and the companies ignoring the threat. Or worse: they threaten the person who reported the vulnerability. An excellent example is Michael Lynn and "Ciscogate" (2005).[16] Lynn informed Cisco of a vulnerability and Cisco ignored the problem. When Lynn decided to make it public, Cisco threatened to sue Lynn and, at Cisco's urging, Lynn's employer forced him to resign. Similarly, back in 2001, a security researcher was arrested for divulging details of an Adobe vulnerability.[17] These are only two of many examples; Attrition.org maintains a record of prominant examples where security researchers were sued for reporting on vulnerabilities over the last 15 years.[18]

   Granted, many times the security reporter has a different view on the severity of the exploit. However, if the company is willing to sue to keep the vulnerability quiet, then the exploit is probably significant. The person reporting on the vulnerability should have immunity from prosecution. This provides an incentive for people and corporations to do the right thing when they learn of an exploit.

5. *Define common standards for personal information.* What type of data should be considered personal information and what data should be considered sensitive?

   Public information, such as your mother's maiden name or the school you attended, do not offer any means to validate users. Any attacker can typically identifying this public information with very little effort. Yet many sites still ask for public information as an alternate form of validation.

   Some sites store credit card information -- they may show users the first four digits or last four digits as an identifier. Other services use these for authentication. If one service uses the last four digits as an identifier (e.g., Amazon.com displays stored credit cards using the last four digits) and another service uses the last four digits as authentication (e.g., GoDaddy.com asks "can you verify the last four digits of the credit card you used?"), then a disclosure at one service can be used to compromise a different service.

   Services need to agree on what data can be used as identification and what can be used as authentication. Otherwise, there is always the risk of a compromise from cross-service information.

6. *Bring together industry and lawmakers*. After industry has identified standards, best practices, and acceptable policies, then we need someone to facilitate cooperation with lawmakers. The current state of large exploits driving laws results in laws that are

[16] http://archive.wired.com/science/discoveries/news/2005/08/68435?currentPage=all
[17] http://news.cnet.com/2100-1001-270082.html
[18] http://attrition.org/errata/legal_threats/

ineffective, unenforceable, and/or outdated. Moreover, many of the new online legislative rulings are driven by special interests that carry more influence than the technical industry. Public forums marshalled by the Department of Commerce seems an appropriate way to first bring together a disjointed industry and then to bring the industry's consensus to Congress.

Network and Infrastructure Security

Malware Mitigation

Computer viruses, spyware, and other forms of malware exploit vulnerabilities. Antivirus technologies attempt to identify known malware signatures at or after the system has been compromised. Unfortunately, it is too easy for virus writers to alter their code in order to avoid signature-based detection. As antivirus vendor Trend Micro's CEO Eva Chen said in 2008, "I've been feeling that the antivirus industry sucks. If you have 5.5 million new viruses out there how can you claim this industry is doing the right job?"[19]

Mitigating the effectiveness of malware does not need to be signature based. Virtual machines, sandboxes, and restrictive access can be very effective at mitigating this risk. In addition, there are known security-oriented best-practices that can easily mitigate most malware risks but are often ignored by developers and system administrators.[20] For example:

- *Modularity and compartmentalization*: Each device, function, and protocol should perform a single task. This allows network issues to remain isolated and distinct. As an example, extremely sensitive information should not be stored on the same network as harmless public data. A lack of modularity led to data comprises at Best Buy (2003),[21] CardSystems Solutions (2005),[22] TJX (2007),[23] Target (2013),[24] as well as many others.

- *Least privilege*. Applications and protocols should operate at the lowest privilege level possible.

- *Controlled environment*. Applications operate under specific user privileges. Applications should not span privilege levels or user accounts.

- *Define trust*. Most protocols operate independently of each other. As a result, a security issue found in one protocol remains limited to that protocol. In contrast, most

[19] http://www.channelregister.co.uk/2008/06/22/trend_micro_eva_chen/
[20] Krawetz, N. (2006) *Introduction to Network Security*. Charles River Media. ISBN 1-58450-464-1.
[21] W1nt3rmut3, "Best Buy Insecurities", *2600 The Hacker Quarterly*, Vol. 20.1, Spring 2003, p. 21-22.
[22] http://www.csoonline.com/article/2130877/data-protection/data-protection-the-15-worst-data-security-breaches-of-the-21st-century.html
[23] http://www.wired.com/2009/07/pci/
[24] https://corporate.target.com/about/shopping-experience/payment-card-issue-faq

applications assume that some dependent protocol will perform authentication, validation, and other security checks. This assumption establishes a false level of trust between the network traffic and application. Applications typically assume that the information has not been intercepted or altered, but no steps are taken to validate this assumption; most network connections operate with little or no explicit security precautions.

- *Validate inputs*. If an input should be a number, then functions should verify that it is number. If the input should be text that is 20 characters long, then validate that it is text that is 20 characters long. Input supplied by an outside source, whether it is an external function or a remote web client, must be validated; applications must not assume that external information is “safe”. Exploits from cross-site scripting, buffer overflows, database injection, and other common attacks are virtually always due to unchecked inputs and the incorrect assumption that external data is trusted.

- *Valid outputs*. With network traffic, corporations and online services should only permit known traffic to exit the system. For example, if a server should never send email, then egress filtering will identify and block any attempts to send email. If a system is compromised with spam software, then the spam software will be ineffective. Similarly, if a server does not use FTP, then FTP should be disabled. In general, the easiest approach is to deny by default and explicitly permit only the necessary protocols.

- *Error handling*. What happens when a security condition cannot be validated? With HTTPS (SSL and TLS), most mobile browsers will accept trusted certificates if the certificate authority cannot be reached. At best, these applications prompt the user -- as if the the user knows how to manually validate a certificate (which is almost never the case). The correct response is to not leave the choice up to the user. Otherwise, the application allows a non-technical user to potentially compromise the system security. With most users, their desire to access a web site is stronger than their knowledge of the potential risks.

  As another example, RFC791 defines network ICMP packets.[25] This networking standard does not define how to handle invalid checksum values. As a result, invalid packets may be accepted, rejected, or dropped -- depending on the implementation.

  Monitoring logs for alerts and suspicious behavior also falls under error handling. According to public reports, the 2013 data breach at Target included alerts related to the compromise, but the alerts were ignored by the company.[26]

[25] http://www.rfc-base.org/txt/rfc-791.txt
[26] http://www.pcmag.com/article2/0,2817,2454977,00.asp

- *Defense-in-depth*. One security mechanism protects one attack vector. Each security precaution adds more complexity for the attacker. By stacking security checks at all levels, it makes the ability to successfully compromise a site significantly more difficult.

Successfully implementing a security strategy takes planning and effort. However, this effort should result in minimized long-term maintenance issues.

As an explicit example of good cybersecurity practices leading to less malware, consider the Agobot malware from 2003-2008.[27] Agobot (also called Phatbot and Gaobot) was described as "a virtual Swiss Army knife of attack software".[28] However, even without antivirus signatures, common network security practices easily stopped this malware. For example:

- Agobot used a remote control system that depended on IRC (internet relay chat) and P2P protocols. Disabling IRC or P2P via egress filtering prevented this remote control system.

- After infecting a system, Agobot would benchmark the network connection. Requiring authenticated outbound traffic, such as a corporate SOCKS5 server, prevented the benchmarking.

- Agobot provided spam support. Limiting outbound email to an authenticated smarthost [29] deters spamming. Today, most large ISPs require use of an smarthost for sending email. For example, Comcast users must relay email through Comcast's outbound mail server, and this server requires authentication in order to relay email. Comcast does not allow subscribers to send email directly.[30]

- Agobot would scan the local network for other hosts to infect. The use of intrusion detection systems (IDS), intrusion protection systems (IPS), and honeypot systems would detect and mitigate the ability for this malware to spread.

- Agobot would capture all traffic on the network (an activity called *promiscuous network monitoring* or *promiscuous mode*). A simple network architecture, such as deploying local network switches or using a star network configuration, limits the amount of data that a promiscuous host can collect.

- Agobot could capture packets and issue redirect commands, sending internal network traffic through compromised network systems. The same steps used to mitigate promiscuous mode also reduce this risk.

[27] http://www.washingtonpost.com/wp-dyn/articles/A447-2004Mar17.html and http://archive.wired.com/techbiz/it/news/2004/05/63393
[28] http://www.dslreports.com/shownews/40854
[29] http://en.wikipedia.org/wiki/Smart_host
[30] Sending email indirectly, through services such as Gmail and Yahoo! Mail, is permitted.

- Agobot was designed to issue a variety of network-based denial of service attacks. Between tuning network protocol parameters, IDS/IPS systems, and egress filtering, the impact from any attack issued from an infected host becomes negated and easily detectable.

In effect, basic cybersecurity practices would mitigate all of Agobot's features. These steps would reduce the likelihood of an infection and limit the impact of any infection. Antivirus signatures are not essential if the core functionality from the malware is addressed by appropriate network security options.

Unfortunately, most ISPs, corporations, and users do not follow basic cybersecurity procedures. This resulted in hundreds of millions of infected computers.[31] In the case of users, they probably do not know how to secure their networks or what best practices to follow. With ISPs and corporations, the lack of applied best practices are typically due to a combination of laziness, insufficient staff, and apathy.[32]

Web Security and Consumer Trust

Web Security

There is a widespread effort to enable HTTPS (TLS or SSL) on web sites for security. However, SSL is typically used with server-side, and not client-side, certificates. Without client-side certificates, the initial HTTPS connection becomes vulnerable to hijacking. Organizations need to agree on a secure method for distributing client-side certificates. Similarly, browsers must make it easier to install both temporary and permanent client-side certificates.

JavaScript is commonly used to make websites interactive. However, there are no Document Object Model (DOM) functional hooks for querying the HTTPS certificate information from browser languages such as JavaScript. The DOM must be extended so that client-side applications can validate the HTTPS connection.

A significant limiting factor for HTTPS adoption is the price. Server-side certificates are typically costly, running up to hundreds of dollars per year. The cost becomes a prohibitive factor for widespread adoption. Moreover, widespread SSL compromises, such as Heartbleed and POODLE, forced many web sites to pay an additional fee for changing their server-side certificates.

[31] http://archive.wired.com/techbiz/it/news/2004/05/63393
[32] http://www.hackerfactor.com/blog/index.php?/archives/638-Security-By-Apathy.html

Organizations such as Mozilla, Akamai, Cisco, and the Electronic Frontier Foundation (EFF) have announced sponsorship of a free Certificate Authority.[33] However, this free service is not yet available and only time will determine if this endeavor will become successful. Other free services either exclude commercial use (deterring commercial adoption) or only issue temporary certificates, which increases the burden on ongoing maintenance.

Another option is the use of self-signed certificates. These are free and not limited by a cost factor. Unfortunately, most web browsers flag these as untrusted. And without client-side certificates, self-signed certificates are also vulnerable to initial connection hijacking.

If the DOM were extended to permit HTTPS access from JavaScript, then a third-party certificate authority would not be required to validate the connection. The server would transmit custom code to the client and issue temporary client-side certificates to deter man-in-the-middle attacks.[34]

It is important to recognize that HTTPS does not validate the online service; it only validates the connection. A browser using HTTPS may still be connecting to a hostile server, and the server may still be attacked by a hostile web client. However, the network connection between the client and the server becomes more secure.

Trusted Downloads

Tools exist to perform file hash checks or comparisons. However, there are no standards for supplying checksums for comparison. Moreover, outside of vendor-specific application marketplaces, there is no central repository for storing authenticated signatures.

Currently, operating systems such as Windows 7, Windows 8, and Mac OS X, alert the user that they are about to run software that was downloaded. (Linux and Android do not offer this functionality.) In addition, browsers such as Chrome, Firefox, and Internet Explorer warn users prior to running downloaded applications. However, none of these system authenticate file checksums prior to running the code because there is no mechanism to supply a file signature.

Web protocols need to be adapted to securely transmit signatures for downloaded files. Browsers need to integrate signature support so that downloads can be automatically checked prior to identifying the download as successful. If possible, the signature should also be checked against a known-trusted repository of download signatures.

[33] https://letsencrypt.org/

[34] While the first-time connection to a server would still be vulnerable to man-in-the-middle attacks, subsequent connections would be more secure. In addition, sites could customize their client-side validation JavaScript to mitigate the risk from generic hijacking attempts.

The signature for downloaded files should be more than a simple MD5 or SHA1 checksum. The signatures should be cryptographically signed and identify who supplied the downloaded file's contents and an external third party should be available to validate the signature.

Beyond creating the protocol and implementing it in browsers and operating systems, developers must be allowed to easily generate signatures. In effect, this means that signature generation must be downloadable, easy, and free. Any requirements to upload developer code for signature generation, use vendor-specific proprietary signatures, issue complex steps to make signatures, or pay a fee for generating a signature will prevent widespread deployment.

Trusted Uploads

Along with trusted downloads are issues with trusted uploads. For example, people who want to spread malware and child pornography frequently attempt to upload these files to servers that can receive uploaded files. At best, the hostile and potentially illegal content is rejected by the hosting provider. At worst, the content is not evaluated by the service and simply distributed to visitors.

Unfortunately, there are few systems that will allow the detection of hostile uploaded content:

- Many services do not validate content. They receive uploads and distribute it.

- A few services run antivirus software on uploaded files, but otherwise do not evaluate code. Unfortunately, these services almost never identify the antivirus software used and do not mention that few antivirus scanners catch more than 45% of malware.[35]

- A few services ignore unknown content. For example, if you should only upload text documents and you upload an executable (malware), then the executable will be discarded.

- In the case of pictures and videos, solutions such as Microsoft's PhotoDNA (a perceptual hash system) are proprietary, patent restricted, and seldom shared with any but the largest online services. Moreover, organizations such as NCMEC (the National Center for Missing and Exploited Children -- an organization mandated by Congress for handling reports of child pornography) and ICMEC (the International version of NCMEC) do not appear to have any interest in adopting alternate perceptual hashing systems.[36]

[35] http://dottech.org/157355/symantec-admits-anti-virus-software-is-no-longer-effective-at-stoping-virus-attacks/

[36] In response to inquiries from Hacker Factor (2013-2014), NCMEC repeatedly chose to not respond, while representatives from ICMEC were explicitly opposed to evaluating any solution that was not Microsoft's PhotoDNA -- even when shown that the PhotoDNA perceptual hash was potentially reversible into a picture.

Cybersecurity and the Internet of Things

Network addresses for Internet devices have historically been issued using version 4 of the Internet Protocol (IP or IPv4). However, the IPv4 address space is relatively small and has been completely allocated. For example, AFNIC (Africa) exhausted their allocated network addresses in 2010,[37] APNIC (Asia-Pacific) in 2011,[38] and RIPE (Europe) in 2012.[39] This address space exhaustion impacts users, corporations, and countries. Under IPv4, existing companies cannot easily expand their online presence and new companies cannot establish a foothold on the Internet without addresses. Moreover, the Internet of Things (IoT) cannot rely on IPv4 since there are not enough available network addresses.

This exhaustion of IPv4 addresses was an active concern in 1992.[40] In 1995, a replacement protocol called IP version 6 (IPv6) was introduced.[41] IPv6 permits significantly more network addresses and is ideally suited for IoT; IPv6 is the current basis for IoT products.

IPv6 has been around over two decades and includes very well-defined specifications.[42] However, widespread adoption is still an ongoing process. Google currently estimates that approximately only 7% of users natively use IPv6, but the rate is continually increasing.[43]

Cybersecurity is a significant issue with regards to IoT. One critical issue is that IPv6 has not been properly implemented by most vendors. Virtually everyone from ISPs to router manufacturers have failed to properly implement the entire IPv6 protocol stack. In most cases, vendors only supply a subset of IPv6, with a negative impact on network security. In other cases, providers appear to have intentionally altered their implementation of some IPv6 functionality in order to maintain retain a proprietary foothold.[44]

This widespread lack of compliance opens the way for significant vulnerabilities. Where consumers see internet-enabled refrigerators and coffee makers, attackers see vulnerable computer systems that can be used to stage attacks, route anonymous traffic, and spy on consumers. Full compliance is essential for properly securing networks.

While IPv6 includes some security options, many additional features cannot be deployed due to patent restrictions. Many vendors are patenting viable security options, or patenting around

[37] http://www.afrinic.net/en/statistics/ipv4-exhaustion
[38] https://www.apnic.net/community/ipv4-exhaustion/ipv4-exhaustion-details
[39] https://www.ripe.net/publications/ipv6-info-centre/about-ipv6/ipv4-exhaustion
[40] http://tools.ietf.org/html/rfc1338
[41] http://tools.ietf.org/html/rfc1883
[42] http://www.ipv6.com/articles/general/timeline-of-ipv6.htm
[43] https://www.google.com/intl/en/ipv6/statistics.html
[44] http://en.wikipedia.org/wiki/Teredo_tunneling#Limitations

elements required for adding security options.[45] For example, Patent US 7958220 discusses IPv6 address acquisition and includes security implications, Patent US 1648134 IPv6 encapsulates auto-configuration and authentication, and Patent EP 1641192 restricts IPv6 neighbor discovery, which is essential for full IPv6 functionality. The current software patent system is widely used to restrict security features. Until this land-grab for patenting technologies is addressed, consumers will be limited in their cybersecurity options.

Maintenance is another key issue with IoT. The Internet consists of many obsolete technologies. For example, Microsoft repeatedly tried to obsolete Windows XP, and finally dropped support in 2014.[46] However, according to NetMarketShare,[47] systems running Windows XP *still* account for over 16% of computers online. A major problem is that newer operating systems require newer computer hardware. The cost to upgrade hardware, including upgrading any legacy applications, makes it more affordable and desirable to continue running obsolete equipment. Similarly, many smartphones cannot be easily updated; they are considered disposable technologies; users typically buy a new phone rather than updating an existing device,[48] and that is assuming that upgrading is a viable option.[49] Efforts are being made by carriers to keep older devices in service for longer duration[50] even though vendors end their product support. This issue with legacy devices will also become a problem with IoT. Unless steps are taken today to outline upgrade paths, IoT will result in the widespread use of old, unpatched, and unsupported technologies that will likely pose significant risks to the network and personal privacy.

As the IoT evolves, vendors should look to develop and adopt open standards to support the basic messaging protocols that their industries need. Current examples of this can be found at the OASIS[51] and in the work being performed by the MQTT, SAML, XACML, and KMIP standards. Developing applications around open protocols can result in a more secure system architecture and a more secure IoT.

Internet of Things and Bloatware

Preloaded applications and services, commonly called *bloatware*, pose a significant risk for IoT. Vendors and manufacturers have an incentive to include these features. Desktop computers, laptops, tablets, and smartphones typically come preloaded with applications. As noted in *The New York Times*,[52] "Software companies pay hundreds of millions of dollars to

[45] http://www.researchgate.net/profile/Gianluigi_Ferrari/publication/258626750_Patents_on_IPv6-Related_Technologies/links/0deec5363a62a954f2000000.pdf

[46] https://www.microsoft.com/en-us/windows/enterprise/end-of-support.aspx

[47] http://www.netmarketshare.com/ April 2015, distribution by Operating System Version.

[48] http://www.phonearena.com/news/Americans-replace-their-cell-phones-every-2-years-Finns--every-six-a-study-claims_id20255

[49] http://www.howtogeek.com/129273/why-your-android-phone-isnt-getting-operating-system-updates-and-what-you-can-do-about-it/

[50] http://www.cnet.com/news/how-your-out-of-date-unsexy-smartphone-can-save-you-money/

[51] https://www.oasis-open.org/

[52] http://www.nytimes.com/2008/08/28/technology/28software.html

PC makers like Hewlett-Packard to install their photo tools, financial programs and other products, usually with some tie-in to a paid service or upgrade." Some applications require a user to start them, while others run automatically, regardless of whether the consumer uses the service.

Undesirable and unused bloatware typically accounts for a significant drain on the available battery life.[53] They can also pose significant security risks to the device.[54] On many mobile devices, these pre-loaded trial versions of software cannot be deleted from the device and cannot be disabled.[55] When installing an application from iTunes or the Android Store, users can view the necessary access privileges and choose to install the software. In contrast, preloaded applications offer the user no choice and no means to identify the access requirements.

The IoT offers vendors more opportunity to preload devices with undesirable applications. This will impact usability, device reliability, and system security. Regulations and processes should be made to standardize the ability to remove bloatware and limit how and when these applications are executed.

Business Processes and Enabling Markets

Vulnerability Disclosure

When users access web sites, they may encounter bugs or potential vulnerabilities. Most users either ignore the issue or come back later, hoping that the bug has been resolved. However, it is not uncommon to see security experts, software developers, and amateur curiosity seekers attempt to find the source of the problem. These computer specialists may be interested in identifying the cause, determining the impact to both themselves and their fellow site users, or driven by some other motivation. While some people are interested in compromising the site, many are only interested in having the site work better.

When a site defect is identified, many security researchers attempt to report vulnerabilities to the affected companies. On rare occasions, the site graciously accepts the report and makes a significant effort to resolve the issue quickly. More often than not, the vendors ignore the reports. Automobile manufacturers spent years ignoring reports about "car hacking".[56] The airline industry actively ignored reports about "airplane hacking".[57] Unfortunately, some

[53] http://www.androidauthority.com/worst-performance-sapping-apps-564689/

[54] https://www.techdirt.com/articles/20140423/15401627009/stupidity-installing-bloatware-that-no-one-uses-everyone-hates.shtml

[55] http://www.wired.com/2010/07/bloatware-android-phones/

[56] Examples include http://www.theregister.co.uk/2005/08/02/car_whisperer/ (2005) and http://www.csmonitor.com/USA/2010/0813/Scientists-hack-into-cars-computers-control-brakes-engine (2010).

[57] Examples include http://archive.wired.com/politics/security/news/2008/01/dreamliner_security (2008) and http://gizmodo.com/5452101/the-danger-of-hackers-getting-into-airplanes-flight-computers (2010).

companies respond by suing the person who reported the vulnerability rather than fixing the problem or taking steps to mitigate the risks.

After giving vendors adequate time to respond, it is common practice for the exploit to be presented publicly -- in forums or at conferences. The hope, in these instances, is that public pressure will force companies to take security vulnerabilities seriously. The presentations also act as a notice to the public regarding the type of risks they currently face.

As an example, in 2008 three students decided to present on subway vulnerabilities after the Massachusetts Bay Transit Authority failed to respond to the vulnerability report. The MBTA responded by suing the three students rather than addressing the security risks.[58] In 2011, Apple decided to revoke a developer's license after security researcher Charlie Miller discovered a vulnerability in the iOS platform.[59] And in 2012, Facebook sued security researcher Glenn Mangham after he reported a vulnerability.[60] Mangham was sentenced to jail time.

In lieu of legal threats, potential jail time, and exacerbation from unresponsiveness, some security researchers have chosen to make vulnerabilities public without giving vendors prior notice.[61]

Although some vendors do welcome vulnerability reports, then are an extreme minority. A few vendors have even initiated "bug bounty" programs, where they reward researchers for responsibly disclosing vulnerabilities. Microsoft,[62] Facebook,[63] and Google[64] all have bug bounty programs. These programs reward users who identify bugs and security risks and who practice responsible disclosure.

Unfortunately, some companies have started bug bounty problems but clearly do not understand the purpose. For example, earlier this month United Airlines announced the first bug bounty program among the airline industry.[65] (They reward in airlines miles rather than US dollars, but many security researchers are happy with a "thank you" or a t-shirt.) The problem with United Airlines is that their bug bounty program places too many restrictions and explicitly threatens to sue rather than be grateful for any reports.[66] For example:

[58] http://tech.mit.edu/V128/N31/subway.html and http://www.cnet.com/news/judge-orders-halt-to-defcon-speech-on-subway-card-hacking/

[59] https://www.techdirt.com/blog/wireless/articles/20111107/18193216671/find-vulnerability-apple-software-lose-your-license-as-apple-developer.shtml

[60] https://nakedsecurity.sophos.com/2012/02/20/jail-facebook-ethical-hacker/

[61] Examples: http://seclists.org/fulldisclosure/2012/Jul/49 and http://www.gossamer-threads.com/lists/fulldisc/full-disclosure/10238#10238.

[62] https://technet.microsoft.com/en-us/security/dn469163.aspx

[63] https://www.facebook.com/whitehat

[64] http://www.google.com/about/appsecurity/reward-program/

[65] http://www.united.com/web/en-US/content/Contact/bugbounty.aspx

[66] http://www.hackerfactor.com/blog/index.php?/archives/674-The-Friendly-Skies.html

- United Airlines states that they are interested in learning about brute-force exploits. However, they also say that any attempt to brute-force information on their system will "result in permanent disqualification from the bug bounty program and possible criminal and/or legal investigation." In other words, they claim to want people to submit exploits, but any attempt to verify the exploit prior to submission could result in legal action.

- United Airlines wants information about "Timing attacks that prove the existence of a private repository, user or reservation". However, they also state that compromising or testing of this type of exploit will result in possible criminal charges.

- United Airlines wants information about cross-site scripting, cross-site forgery, and bugs on customer-facing web sites. However, cross-site attacks are a form of code injection, and any code injection on live systems is forbidden. Moreover, all customer-facing web sites provided by United Airlines are explicitly live systems.

- United Airlines includes a list of "Bugs that are not eligible for submission". These include bugs related to the "onboard Wi-Fi, entertainment systems or avionics". While we agree that testing on these systems places a flight in danger, the exclusion explicitly states that they are not interested in learning about vulnerabilities that could endanger lives. It is as if United Airlines would rather have passengers die than address a computer security issue. United Airlines also ignores the fact that malicious users (bad guys) have no incentive to follow these arbitrary restrictions.

- When submitting a report to the United Airlines bug bounty program, they respond with an automated confirmation of receipt. However, they do not provide any form of tracking identifier. This means that there are no mechanisms for supplying additional information from the reporter, associating any follow-up from United Airlines, or tracking the bug report's status.

What we, as an industry, need are:

1. Clear guidelines for defining responsible disclosure,
2. Clearly defined steps in the event that a company ignores the reporting,
3. Acceptable ways to acknowledge a report.
4. Acceptable methods to follow-up and check the status of a report.

Moreover, these guidelines must be written in plain English (and not legalese since most software engineers are not attorneys) and made freely available. The Electronic Frontier Foundation (EFF) has taken steps to address many of these issues with their "Coders' Rights Project".[67] However, the EFF's guidelines have not been widely accepted by industry.

[67] https://www.eff.org/issues/coders/vulnerability-reporting-faq and https://www.eff.org/issues/coders.

Other options include standards like ISO 27035 (Security Incident Management), ISO 29147 (Vulnerability Disclosure), and ISO 30111 (Vulnerability Handling Processes). Unfortunately the ISO standards are not free -- there is a fee to view each standard. These standards also have very technical contents. Both the fees and the overly technical content deter widespread adoption.

Enabling Participation

Promoting Participation

A major hurdle to the participation problem is figuring out how to contact all of the stakeholders. Announcing this RFC in the Federal Register may be the officially sanctioned method, but most people in the computer security field do not subscribe to or monitor requests published in the Federal Register. (We did not learn about this request until it was propagated on Twitter.) The Federal Register was first published in 1936,[68] so it predates the Internet and modern technologies. And while the Federal Register does have a web site (federalregister.gov) and a blog (updated infrequently), it has no apparent presence on Twitter, Facebook, or other social networks where it is likely to reach the technical community.

This particular RFC announcement from the Department of Commerce was mentioned on Twitter[69] on March 20 by the RFC's author. However, as with most things on Twitter, this announcement was not distributed to a wider audience until subsequent mentionings.[70]

There are many ways to reach computer security subject matter experts. Social media, such as Twitter and Facebook, are one option. However, this field has conferences, social meetups, and security-oriented groups that meet every few days, all over the nation. Most weeks, there are at least two computer security conferences somewhere in the United States. These meetings range from large, sponsored conferences to training seminars and small social gatherings.

It is not necessary to attend all of these groups, or even all of the large conferences. If something valuable is presented at one meeting, then word will spread socially to most people in this field. With more meetings (or larger groups), word will spread faster. In addition, there are many public online forums, such as the Full Disclosure mailing list, and announcement forums, such as the Internet Storm Center, where information can be rapidly disseminated. Finally, there are many popular security feeds where, if the content is mentioned, then it will reach a vast number of people. (Bruce Schneier's Cryptogram newsletter, the CERT Podcast series, and Tenable Network Security's podcast are all potential candidates -- *if* their respective hosts are interested in discussing the topic.)

[68] http://www.gpo.gov/fdsys/pkg/USCODE-2011-title44/html/USCODE-2011-title44-chap15-sec1505.htm
[69] https://twitter.com/allanfriedman/status/578906314505633792
[70] https://twitter.com/allanfriedman/status/598145505911242753

After getting the word out for feedback, the issue becomes coordinating a physical meeting for this discussion. Large conferences such as Defcon, RSA, and Shmoocon are one such option. Another option are the B-sides conference series (conferences that parallels some of the larger conferences). Alternately, you may consider holding 2-3 town hall meetings that are accessible via online participation. Physical locations should be associated with technology areas. Strong candidates include the Virginia/Maryland/DC areas, the California Bay Area (San Jose being a strong choice), or Denver, Colorado (a location centralized for most of the United States). A townhall meeting with online streaming, chat rooms, and open to online participations would offer convenient venues for most of the stakeholders.

Potential Workshops or Events

This DoC RFC asked for recommendations as to whether these cybersecurity issues would be better served by a single workshop or by a multi-stakeholder, consensus-building process.

There are a wide variety of actors with a stake in the strength of cybersecurity. These include:

- *The security community*. These people are typically interested in having the strongest security options available.

- *Academia*. These people are actively researching next generation technologies. Their solutions may appear theoretical today. However, the choices made today will significantly impact the ability to deploy future technologies.

- *Government*. This group includes military, intelligence, and law enforcement, both domestic and partner nation states. In general, they want to the strongest solution for their internal networks, but a weaker solution for the general public and anyone else. We frequently see Congress propose laws that would enforce weaker security. For example, from 1992 to 2000, the United States had very strict export restrictions around cryptographic algorithms.[71] Even today, there are some restrictions around the availability of cybersecurity solutions -- these are regulated by the Department of Commerce.[72]

- *Developers*. Outside of the security community, developers typically want to easiest solution. Unless they have a reason to implement security mechanisms, most developers will not make any extra effort to implement any form of cybersecurity. In most cases, developers outside of the security community have either no insight or a vague notion of security-oriented best practices.

[71] http://en.wikipedia.org/wiki/Export_of_cryptography_from_the_United_States
[72] https://www.bis.doc.gov/index.php/forms-documents/doc_view/335-supplement-no-1-to-part-774-category-5-part-ii-information-security

- *Consumers*. There is a common belief that consumers do not know what they want.[73] In general, users are interested in usability and viability for a specific task, but they often seek the wrong solution for specific problems. Consumers typically do not care about cybersecurity until a compromise impacts someone they know. Implementations around network security and personal information protection should be transparent to the user.

- *Consumer Advocates*. In contrast to consumers, consumer advocate groups represent the best interest of consumers. These groups typically want choice for solutions, transparency for implementations, and privacy for personal information.

- *Corporations*. A minority of corporations treat security as a primary requirement. Instead, companies are often marketing-driven. They just want to check off "encryption" or "security" on their feature list, without any thought as to the completeness of the solution.

  A good example of this is the Hewlett-Packard Secure Web Console. The "security" in the Secure Web Console came from performing a simple bit-flip operation in a fixed pattern (XOR with 0x37) to the network traffic. In effect, the security was nothing more than an item on a checklist and not an effective security solution. As Michael Shaffer noted in November 2000,[74] "the HP Secure Web Console is not likely to provide a sufficient level of security from any but the most naive attackers".

  Without regulations requiring the protection of personal information or harsh penalties for violations, corporations have little incentive to treat cybersecurity as something other than a checklist item.

  Some corporations view policies and regulations as a means to create an unbalanced marketspace. Through patents, lobbyists, and corporate pressure, they can manipulation regulations and requirements in order to benefit their own corporate goals. For example, the original IPv6 specifications had a mandatory requirement for IPsec (an end-to-end security protocol). IPsec is a secure architecture proposed by BBN Technologies[75] and uses standards drafted by Microsoft, Checkpoint, Motorola, MIT, NIST, independent contributors, and other corporations.[76] Having IPsec as a mandatory requirement ensures that all IPv6 connections have the option to use a secure VPN connection, and VPN security implementations are equivalent and compatible. IPsec does not rule out other security options; it only provides a common baseline for the implementation.

[73] http://www.forbes.com/sites/chunkamui/2011/10/17/five-dangerous-lessons-to-learn-from-steve-jobs/
[74] http://www.giac.org/paper/gsec/172/cracking-hp-secure-web-console/100647
[75] https://tools.ietf.org/html/rfc4301
[76] The specifications cover dozens of RFC documents, including https://tools.ietf.org/html/rfc5996, https://tools.ietf.org/html/rfc4305, and https://tools.ietf.org/html/rfc4307.

In December 2011, a new specification called RFC6434 changed IPsec from *required* to *optional* (see RFC6434 section 11).[77] This revision was provided by authors who represent SRI International and IBM Corporation.[78] By removing the IPsec requirement, they remove any common basis for secure VPN implementations. This lowers the availability of a secure solution for consumers and increases the likelihood of vendor-specific and incompatible solutions.

In contrast to consumer advocates, corporations typically view consumer information as an asset.[79] It is information that can be collected, sold, and used for a commercial profit. As European Consumer Commissioner Meglena Kuneva noted in 2009, "Personal data is the new oil of the Internet and the new currency of the digital world." [80] Because personal data requires little or no proprietary technologies to collect, there is little incentive to protect it.

Since each of these groups have their own agendas, it will likely be easiest to hold separate events for the different groups, and then an overall event that brings together the key findings from each set of participants. Holding a single meeting with all groups present will likely lead to corporate and government interests dominating opinions from other stakeholders. In addition, the wide variety of views, opinions, and private agendas will likely turn any single, large meeting into a no-holds-barred brawl with the loudest voices drowning out other opinions.

Contributors

This RFC response is provided by Dr. Neal Krawetz of Hacker Factor, with feedback from Joseph S. Klein of Disrupt6 and an anonymous reviewer.

If you have questions, please contact:

Neal Krawetz, Ph.D.
Hacker Factor Solutions
PO Box 270033
Fort Collins, CO 80527-0033
http://www.hackerfactor.com/

[77] https://tools.ietf.org/html/rfc6434
[78] A third author, representing Nokia, is present on both draft standards, including and excluding IPsec from IPv6.
[79] http://www3.weforum.org/docs/WEF_ITTC_PersonalDataNewAsset_Report_2011.pdf
[80] http://europa.eu/rapid/press-release_SPEECH-09-156_en.htm