The Internet is Broken: Idealistic Ideas for Building a ~~NEW~~GNU Network

Christian Grothoff Bartlomiej Polot Carlo von Loesch

The GNUnet Project

1 Introduction

The Internet is broken, by design. Recent revelations about the abuses by the NSA or GHCQ rarely contain stunning details about new magical technical capabilities of these agencies, but instead merely detail that they have both the budget and the moral framework to exploit known vulnerabilities at scale.

The problems of today's Internet start at the Ethernet layer, where sender's MAC-48 addresses can be faked, and packets can be intercepted and modified by switches. The TCP/IP layer has the same problems, with routers learning source and destination of all communications, as well as details about the payload (such as port numbers). Routers can also interfere with connections, for example by injecting RST packets. TCP also operates with the assumption that other traffic will be "TCP-friendly", which is like having a speed limit on roads without enforcement. TLS, the workhorse for today's "Internet security", provides "security" only if all of hundreds of certificate authorities operate correctly (which they usually do not), and it comes with a large set of supported cryptographic primitives, most of which are known to be insecure.

All of the above facts are well-known and even discussed in ordinary news venues *before* Edward Snowden decided to expose some of the systemic abuses supported by these design flaws.

We will present ideas for fixing the network, with the goal of building a GNU[1] network, where users have the freedom (0) to securely access information ("run" the network), (1) the freedom to study all aspects of the network's operation ("access the code"), (2) the freedom to distribute information ("copy"), as well as (3) the freedom to deploy new applications ("modify"). Today, monitoring infrastructure, proprietary implementations, traffic shapers and firewalls restrict all of these essential freedoms to some degree.

2 Addressing

"Can we obfuscate metadata with less overhead than Tor?"

Addresses on the Ethernet and IP layers today are controlled by manufacturers and Internet Service Provicers. They offer no security and are only moderately useful for routing (considering the growth in BGP routing tables)[2]. Furthermore, in both layers each packet includes the sender address in cleartext,

[1] `http://www.gnu.org/philosophy/`

[2] `http://bgp.potaroo.net/`

which leaks information to surveillance devices deployed in the network and has prompted the need to develop anonymization software like Tor[3].

We envision a radical departure from this approach. Instead, frames and packets should be routed using an (ephemeral) public key to identify the destination. On the data link layer, the public key would identify the device within the network of an organization, and on the inter-networking layer the public key would identify the target organization. The payload, including the sender address information and higher-level protocol identifiers, would be encrypted (and authenticated with a MAC). Layered encryption would be used, such that first the target organization would need to decrypt the packet to determine the ultimate target within the organization, and then the target device would use its private key to obtain the sender's address information and the payload.

Clients would use keys with extremely short lifetimes (as they would only need to announce new keys for routing within their organization) while servers and organizations would use ephemeral keys with modest lifetimes for addressing, achieving so-called perfect forward secrecy. Clients would determine the current ephemeral key(s) for a given service (and its respective organization) using a name system, similar to how hostnames are resolved to IP addresses with DNS today. Using ECC, public key addresses could be as short as 32 bytes, which is comparable to the 16 bytes of IPv6 addresses. A fixed moderate packet size, such as 64k, should be used to further hinder traffic analysis and reduce the number of public-key operations.

While this architecture creates challenges for routing protocols, it addresses a wide range of security issues as protocols such as ARP (spoofing), DHCP (spoofing, resource exhaustion), NDP (spoofing), IP (neutrality), TCP (lack of authenticity, lack of encryption, RST attacks), UDP (lack of authenticity, lack of encryption) are all impossible by design.

3 Routing

"Can lower layer changes (e.g., to IPv6) help?"

BGP uses policy-based routing which requires contract negotiation to establish business policies. Dominant players are in a position to extract transit fees, putting smaller businesses at an inherent disadvantage. The resulting routes still depend on the correct behavior of participants; operators can censor or blackout entire countries, and hijacking foreign traffic is also possible.[4] We need a routing mechanism that will work in an unmanaged, decentralized environment, and where incentives and rewards for proper routing are embedded into the protocol instead of into signed paper documents.

Distributed Hash Tables (DHTs) [3] enable unmanaged and decentralized routing. In a DHT, each node has detailed knowledge of a small (typically $O(\log n)$) portion of the network, and greedy algorithms are used to locate the target node (in typically $O(\log n)$ to $O(\sqrt{n})$ steps). The first generation of

[3] `https://torproject.org/`

[4] `http://bgpmon.net/?p=282`

overlay DHTs assumed universal connectivity in the underlying network: any node in the DHT was expected to be able to talk to any other node in the network directly. Recently, DHTs for restricted-route networks have lifted that restriction [2, 5], allowing DHTs to operate despite physical connectivity being constrained.

It is possible to use a censorship-resistant DHT (where malicious nodes cannot prevent lookups) to determine a set of redundant paths to a target and establish an (encrypted) tunnel via redundant connections. Such a tunnel can then be used to create SCTP-like communication channels. Reputation systems can be used to provide incentives to systems that properly relay traffic, for example by prioritizing requests by reputation and enabling participants to trade reputation for currency.

4 Public Key Infrastructure

"Can we deploy end-to-end crypto?"

The main Public Key Infrastructures (PKIs) on the Internet are hierarchical. DNSSEC uses the hierarchy established by DNS to provide authenticity and X.509 Certificate Authorities (CAs) form a hierarchy rooted in the browser's root store. Neither system offers any kind of trust agility to the user; if any of the high-profile targets in the unique trust chain are broken by technical or legal attacks, all security is lost. An extensive history of attacks against these systems can be found in [4].

The PGP Web-of-Trust (WoT) is a decentralized alternative, but it suffers from unintuitive trust models and its dependency on DNS and SMTP for user@domain addressing. Rivest's SDSI/SPKI [6] provides a starting point for an alternative, fully decentralized public key infrastructure. Here, users certify each other's public keys like in the WoT, but instead of validating that a government-issued identity card is owned by someone who can read e-mail at a particular address, the paths through the signature graph can be used to identify users.

The GNU Name System [7, 9] combines this idea with the use of a DHT to resolve names to values. Cryptography is used to ensure the authenticity of the values and that the lookup and the returned information remains private.

Breaking out of hierarchical name systems requires not only a change in technology, but also in the mindset of users. We must expect a change in usage patterns once identities are no longer underwritten (and thus controlled) by governments or international cooperations.

5 Content Distribution

"How do we reconcile caching with end-to-end encryption?"

Today's main Internet applications, such as the web and social networking applications, use a client-server paradigm. At best they are federated, like

email or instant messaging, but ultimately in most cases the users' data resides on servers not controlled by the users themselves. The advantage for the users is that they do not need to setup their own service, and might have a network presence even if they themselves are offline. The price is that their communication is exposed to third parties; even if the servers were trustworthy, the use of opportunistic caches for performance limits the use of proper encryption. Even if the data were always end-to-end encrypted, the servers can easily observe metadata. Also, as recent revelations have shown, servers are hard to secure and by aggregating data of many users naturally assume the form of pots of honey.

To enable low-bandwidth devices distribute information to a large number of clients without entrusting the data to servers, the system needs to offer a cooperative secure multicast mechanism controlled by the endpoints. As demonstrated by the Tor network, endpoints can enlist a relay infrastructure — where relays have no idea what data they are delivering or for whom — to significantly boost the performance of the low-bandwidth endpoints that are in control. The multicast system can also collaboratively address the problem of information persistance, allowing users that are not online at the same time to still comunicate. We envision a self-organizing system that uses the target group to provide multicasting and persistence. This system will eliminate the need of network caching, as every device will get the data from the multicast group and store it locally.

Servers are also usually the point where protocol update decisions are made and long-term data is migrated from one version of the application to the next. While a centralized application can be updated just modifying the central server, a decentralized application cannot ensure the end users use the last version of the software. Thus, in a decentralized network, an extensible protocol is needed that structures the information. When new features or applications are deployed, the protocol must ensure the software can still make some sense of the semantics of unknown extensions. This is not impossible; the PSYC protocol [8], which is extensible in ways similar to object-oriented languages, offers this capability.

6 Applications

"Can we deploy other applications so that we mitigate monitoring?"

Many popular modern applications on the Internet were developed with a simple primary goal: to turn a profit. The dominant client-server paradigm implies that applications need a server, and servers create operational costs that must be paid. As a result, applications are designed to increase consumption — directly via sales, or indirectly via advertising. However, given the finite nature of the planet, increasing consumption rarely represents positive social change.

Once we fully decentralize the network and enable cooperative hosting and routing of information, the need to create profits to sustain cooperations disappears. Peer-to-peer (P2P) applications can in principle be viable and provide high-quality services to the population without significant investments into in-

frastructure, as the music monopolies had to learn at the turn of the millenium.

The current condition of the Internet makes it a must for business players to collect and market big data in order to be competitive. Even if it is in their best intentions to do no evil, corporations have to play by the rules of the market. The GNU network creates a new level playfield with more equal business opportunities that do not collide with the secrecy of correspondence and other civil rights.

7 Migration

A new global network is unlikely to be developed and deployed overnight, so the presented vision can only become reality with a transition plan. Existing P2P networks have already answered this question. Modern P2P networks generally operate as overlay networks, that is, they implement their own addressing and routing over TCP/IP. Thus, we can begin deploying these systems today without changes to the existing infrastructure. However, as a result, these systems will require complex bandaids to work around problems created by the existing Internet design, which creates security issues and impacts performance.

Thus, in a second step we should begin to replace the existing Internet infrastructure, replacing Ethernet and TCP/IP hardware with network equipment that provides security and privacy also at the lower layers. Users that are connected using these new methods may want to still access the legacy Internet, but this is easily facilitated by tunneling TCP/IP over the GNU network to a peer that acts as a bridge.

8 Conclusion

Today's Internet is a complex piece of systems software that has undergone much patchwork maintenance over many decades. As Brooks famously wrote: for large systems projects one should "plan to throw one away; you will, anyhow" [1], and for the Internet that time has clearly come. Those familiar with the design flaws of the Internet can only be surprised by the *scope* of the revelations about national insecurity agencies using the Internet for mass surveillance and intrusion, but not by the *technical reality* that supported those abuses.

We must seize the opportunity to build a better network. This paper presented ideas and a migration strategy to transition to a radically different design, which addresses some of the key challenges we face today. Most importantly, our design avoids hierarchical structures, "trusted" authorities or central points of failure, as those will be exploited by authoritarian organizations and thus contribute to the collapse of civil society.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) under ENP GR 3688/1-1.

References

[1] F. P. Brooks, Jr. The Mythical Man-month (Anniversary Ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[2] N. Evans and C. Grothoff. R^5N: Randomized Recursive Routing for Restricted-Route Networks. In 5th Int. Conf. on Network and System Security, pages 316–321, 2011.

[3] S. Götz, S. Rieche, and K. Wehrle. Selected DHT Algorithms, volume 3485 of LNCS, chapter 8, pages 95–117. Springer, 2005.

[4] R. Holz. Empirical analysis of Public Key Infrastructures and investigation of improvements. PhD thesis, TU Munich, submitted December 2013.

[5] P. Mittal, M. Caesar, and N. Borisov. X-vine: Secure and pseudonymous routing using social networks. CoRR, abs/1109.0971, 2011.

[6] R. L. Rivest and B. Lampson. SDSI – a simple distributed security infrastructure. `http://groups.csail.mit.edu/cis/sdsi.html`, 1996.

[7] M. Schanzenbach. A Censorship Resistant and Fully Decentralized Replacement for DNS. Master's thesis, Technische Universität München, 2012.

[8] G. X. Toth. Design of a social messaging system using stateful multicast. Master's, University of Amsterdam, Amsterdam, 2013.

[9] M. Wachs, M. Schanzenbach, and C. Grothoff. On the feasibility of a censorship resistant decentralized name system. In 6th International Symposium on Foundations & Practice of Security (FPS 2013), LNCS, page 14. Springer Verlag, 2013.