Mark Donnelly
Sam Hartman
January 15, 2014

Beyond Encrypt Everything: Passive Monitoring

Discussions of pervasive monitoring within the IETF have focused on large-scale passive monitoring of the Internet. The most straight-forward objective in combating pervasive passive monitoring is to encrypt more. This paper explores what happens when we succeed. What will the residual threats look like? Where will the pressure exerted by those who want to monitor the Internet move? What can we do to respond to the new pressure?

We believe that this broader analysis will help as we examine mitigations for pervasive monitoring. In looking at pervasive monitoring we consider the sorts of passive attacks that are being included in the Internet Threat Model [1] as well as more active pervasive actions designed to monitor the Internet.

Together the pervasive passive monitoring as well as the more active monitoring initiatives directly impact confidentiality and privacy. Credentials collected as well as indirect compromises impact integrity and potentially availability of the Internet. We propose the following goals. First, increase the difficulty of monitoring without the knowledge of the parties being monitored. Second, provide users of the Internet better tools to understand their risks of monitoring. Finally, increase the probability that attempts to monitor a given interaction will be detected. These goals are consistent with those discussed at the PERPASS BOF at IETF 89 [2].

Encryption and its Limitations

A lot of initial focus has been on encryption. We've seen that encryption increases the difficulty of an actor attempting to monitor a connection. Without encryption of some kind, Internet traffic is trivially vulnerable to passive monitoring. The questions tend to focus around whether a particular approach to encryption provides significant value in a certain context. The IETF is in the process of describing the interactions between encryption protocols and pervasive monitoring. As an example, Perfect Forward Secrecy (PFS) could prevent an attacker from monitoring a conversation even if the attacker knows the long-term private key used by a server. For TLS-based protocols, Achieving PFS requires selection of a compatible TLS cipher suite as well as properly discarding ephemeral keys. Similarly for some applications, encrypting the conversation even when neither side of the connection is authenticated may provide a significant defense against passive monitoring. However, such opportunistic encryption is vulnerable to a man-in-the-middle attack. In cases such as attacks on signaling protocols such as SIP, mounting large-scale man-in-the-middle attacks may provide significant value to an attacker and may be technically feasible

Off the Record (OTR) [3] is an example of an encryption system with relatively good usability that provides significant defense against passive monitoring. OTR works to provide PFS for chat clients. The encryption is between the two endpoints of the chat and does not rely on encryption at the chat server layer. If the chat server does not mount a man-in-the-middle attack, then the encryption provides confidentiality. Optional authentication facilities can detect man-in-the-middle attacks.

OTR also demonstrates some of the residual risks of encryption. OTR is unable to protect identity; someone who can monitor the connection with the chat server can tell which two parties are chatting even though they cannot detect what is said. Encryption between the client and chat server would provide some additional privacy. However, the IP addresses of involved parties would leak much of the same information.

---

1 http://tools.ietf.org/html/draft-trammell-perpass-ppa-01
2 http://www.ietf.org/proceedings/88/minutes/minutes-88-perpass
3 https://otr.cypherpunks.ca/

In general, statistical analysis of the traffic can reveal significant information about what encrypted resources are transferred [4]. As an example, it is often possible for an attacker to observe traffic and determine whether a particular document is requested even when the traffic is encrypted.

The TOR Project [5] works to combat traffic analysis, hide location and provide anonymous browsing. It accomplishes this at a significant performance and scalability cost. Work on TOR and other anonymity research has shown that many IETF protocols are easy to fingerprint [6] and leak information even when encryption is used. One of the goals of the TLS 1.3 standard is to minimize parts of the exchange that are not encrypted [7]. However, this will not provide a defense against statistical analysis as a way to detect which objects are requested. Even so, encryption can significantly increase the work and decrease the reliability of information collected by those monitoring the Internet.

Certificates and Keys

Especially as more traffic becomes encrypted, the ability to decrypt this traffic becomes valuable to those monitoring the Internet. With TLS cipher suites that do not provide PFS, learning the server's private key provides an attacker the ability to decrypt all traffic sent to or from that server. Using cipher suites that can provide PFS provides a solution. The server can leak its private key (possibly because of legal requirements) but if the server does not leak the private key and changes it frequently, then a purely passive attacker is not expected to be able to decrypt the traffic.

However, an attacker can mount a man-in-the-middle attack. One possibility is to get a certificate issued in the name of the attacked server. Another possibility is to rely on users' willingness to bypass security warnings and go to a site even when the certificate is not valid.

The most successful defense against man-in-the-middle attacks is certificate or public key pinning [8]; this HTTP extension provides continuity between accesses to a given site. If an attacker mounts a man-in-the-middle attack, they need to always mount that attack to a given target or raise a significant risk of detection. As an example, key pinning could have detected an attack in which the US National Security Agency impersonated Google [9] to gather information. Google Chrome includes pinning for most Google sites; this proved instrumental for detecting fake certificates issued by Diginotar [10].

However, if your site doesn't come pinned in the browser distribution, then a man-in-the-middle attack is still possible on first access. Certificate Transparency [11] provides a way for servers to audit which certificates have been issued for their domains. It is hoped that this will significantly increase the chance that an attacker attempting to monitor a site with an improperly issued certificate will be able to do so.

In the past we've proposed using approaches like RFC 5056 to leverage existing relationships with sites [12]. The hope is that existing credentials can be used to detect man-in-the-middle attacks. However that work does not have much in the way of deployment experience and was not designed with pervasive monitoring in mind. It's worth exploring whether that can help detect unwanted monitoring, but the set of trade offs have become more complex.

---

4 http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1004359&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1004359
5 https://www.torproject.org/
6 https://blog.torproject.org/blog/experimental-defense-website-traffic-fingerprinting
7 http://www.ietf.org/proceedings/88/slides/slides-88-tls-4.pdf
8 http://tools.ietf.org/html/draft-ietf-websec-key-pinning-09
9 http://news.cnet.com/8301-13578_3-57602701-38/nsa-disguised-itself-as-google-to-spy-say-reports/
10 https://www.eff.org/deeplinks/2011/08/iranian-man-middle-attack-against-google
11 http://tools.ietf.org/html/rfc6962
12 http://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_9.pdf

Target: JavaScript and Shared Resources

Attackers have engaged in targeted attacks to collect information. Either bugs in code are exploited or alternatively code is modified so that it can be exploited.

Exploitable code is a significant problem whose solution is almost entirely outside of the scope of the IETF and W3C. However standards bodies can play a significant role in making sure that if quality code is available, it is not replaced with code that gives the attacker an advantage.

JavaScript, fonts and other web resources are often hosted on CDNs or central servers rather than being duplicated across websites. This provides caching and performance advantages. However, these sites are potentially very interesting targets for those wishing to monitor the Internet. If the code hosted on these sites is changed, then many systems may run code under the control of an attacker wishing to perform monitoring. Policies such as same-origin [13] will require a bit of cleverness from attackers, but leaking data of interest to the attacker is definitely possible.

Global changes to code might be noticed. However, another possibility is to make targeted changes to the code only for specific clients.

Several mitigations are possible. One is for sites invoking resources from hosted sites to include a hash of the requested resource. This guarantees that the hosting site is not modifying the resource. The subresource integrity specification [14] is a proposal for accomplishing this.

This leaves the problem of how a site knows what hash to include. That is, how do they confirm that the version of the resource they have downloaded to evaluate the hash is globally available and not targeted to them. An approach similar to certificate transparency where public logs of resources are maintained can solve this problem. The proposed Transparency Working Group [15] can provide technology to meet this objective.

A possible attack on the subresource integrity hash is that a government could coerce a site to replace its usual JavaScript reference with a correctly hashed malicious reference. The browser would recognize that the malicious JavaScript hashes to the expected value, but the resource is still undesired. One mitigation might be to track how commonly a given URL references a particular JavaScript resource, though a sophisticated attacker could add entries into a tracking system. Other mitigations should also be explored.

Direct Data Acquisition

One channel for monitoring is for governments to directly approach service providers and demand information. As passive techniques become more difficult this is likely to become a more important channel for monitoring.

The Internet community would be better served by understanding what legal entities are likely to be able to serve process against a service provider. This can be complicated when a service provider uses another service provider—for example when a web site uses a service like Amazon S3 for storage. Other factors can complicate this as well. It would be desirable to have a mechanism for service providers to disclose this information in a programatically parsable manner.

Maintaining Service Provider Diversity

Choice and flexibility is an important cornerstone of the Internet. Large cloud service providers provide added value in exchange for access to users' private information. However smaller, privacy-focused service providers also play an important role. We've seen cases where the failure of privacy-focused businesses served as a warning about government pressure incompatible with

13 https://developer.mozilla.org/en-US/docs/Web/JavaScript/Same_origin_policy_for_JavaScript
14 http://w3c.github.io/webappsec/specs/subresourceintegrity/
15 http://www.ietf.org/mail-archive/web/therightkey/current/msg00680.html

their basic business goals[16].

The IETF and W3C should continue to be committed to developing technical protocols that can be deployed successfully both by large and small providers. This is more than just a platitude. As an example, many approaches to fighting SPAM significantly favor large e-mail providers and make it more challenging to operate a small e-mail provider. Part of fighting pervasive monitoring needs to be looking into technical standards that work well for all sizes of service provider and examining and correcting gaps where the standards make it difficult to operate privacy-focused businesses.

Content Delivery Networks and Templating

Many of the larger websites wish to push significant functionality in the website towards the edge of the network in order to improve performance. CDNs are a fairly common mechanism for accomplishing this. CDNs deliver static content including images, JavaScript and videos. However, CDNs are also involved in using proprietary templating languages to construct pages on the fly to deliver at the edge of the network.

These CDNs pose an interesting target for monitoring on a number of fronts. The edge devices are good points at which to monitor an individual's interaction with multiple websites. The edge devices are also an interesting injection point for exploits. While the CDNs dedicate significant resources to security, securing an edge device against nation-state actors is a significant challenge.

Further, CDNs must employ some technique to localize the user to an edge-server. Whether this is implemented with DNS, anycast, or some other means, localization is key to a CDN's operation. If an agent subverts the localization system and directs the user to an edge-server run by that agent, then the user's traffic is now trivially able to be monitored or controlled.

Standards bodies may be able to help in a number of ways. We could develop operational best practices on how to apply least privilege design, minimizing the impact of an edge server compromise. We could explore whether there are ways in which the content origin could vouch for resources minimizing the impact of an edge-server compromise on Internet integrity.

There may be room for this work both for pure static resources as well as for some template situations. Static resources are easy to vouch; all users of a site receive the same response, and therefore the same vouch. Templating presents a more difficult challenge, since the overall response is unique to each user. However, the overall response is constructed from smaller blocks, many of which are static and therefore easy to vouch. For the fewer areas that are truly dynamic for each user, other means of vouching would have to be explored. In one scenario, the origin could generate a block and a vouch every time the dynamic areas change. In another scenario, the origin could generate a generic vouch for the area, saying that the content will change and should be less trusted. The browser would then deny risky elements, such as scripts or links, and permit a limited set of elements in the area, such as text and whitespace.

Conclusions

We believe that as more traffic becomes encrypted, the Internet will face new challenges in mitigating the risk of pervasive monitoring. Authentication of server and avoiding man-in-the-middle attacks will become more important. Measures to provide a more trustworthy platform and to minimize the number of actors who need to be trusted to maintain integrity and confidentiality are likely to improve privacy. Measures to provide information to users so they can understand their monitoring risk may also play a crucial role.

Many of the necessary measures to address future risks are underway or at least have initial proposals. We recommend continuing to track how all these mitigation strategies interact at a high level to better understand the overall privacy climate.

16 http://lavabit.com/