perpass non-WG
Internet-Draft
Intended status: Informational
Expires: May 17, 2014

B. Trammell
ETH Zurich
D. Borkmann
Red Hat
C. Huitema
Microsoft Corporation

November 13, 2013

A Threat Model for Pervasive Passive Surveillance
draft-trammell-perpass-ppa-01.txt

Abstract

This document elaborates a threat model for pervasive surveillance. We assume an adversary with an interest in indiscriminate eavesdropping that can passively observe network traffic at every layer at every point in the network between the endpoints. It is intended to demonstrate to protocol designers and implementors the observability and inferability of information and metainformation transported over their respective protocols, to assist in the evaluation of the performance of these protocols and the effectiveness of their protection mechanisms under pervasive passive surveillance.

Table of Contents

1. Introduction

Surveillance is defined in [RFC6973], Section 5.1.1, as "the observation or monitoring of an individual's communications or activities". Pervasive passive surveillance in the Internet is the practice of surveillance at widespread observation points, without any particular target in mind at time of surveillance, and without any modification or injection of of network traffic. Pervasive

passive surveillance allows subsequent analysis and inference to be applied to the collected data to achieve surveillance aims on a target to be identified later, or to analyze general communications patterns and/or behaviors without a specified target individual or group.

This differentiates privacy in the face of pervasive surveillance from privacy as addressed in the literature, in that threats to privacy are generally (as in [RFC6973]) taken to have as a specific goal revealing the identity and/or associations of a specified individual; defeating pervasive surveillance of a large population is therefore more difficult than protecting the privacy of a single individual within a larger population.

In this document, we take as given that communications systems should aim to provide privacy guarantees to their users, and that susceptibility to pervasive surveillance should be avoided to the extent possible as a design goal in protocol design. From these assumptions we take the very act of pervasive surveillance to be adversarial by definition.

This document outlines a threat model for an entity performing pervasive passive surveillance, termed the Pervasive Passive Adversary (PPA), and explores how to apply this model to the evaluation of protocols. As the primary threat posed by pervasive surveillance is a threat to the privacy of the parties to a given communication, this document is heavily based on [RFC6973].

2. Terminology

[EDITOR'S NOTE: Check to see whether we actually use these...]

The terms Anonymity, Anonymity Set, Anonymous, Attacker, Eavesdropper, Fingerprint, Fingerprinting, Identifier, Identity, Individual, Initiator, Intermediary, Observer, Pseudonym, Pseudonymity, Pseudonymous, Recipient, and Traffic Analysis are used in this document as defined by Section 3, Terminology, of [RFC6973]. In addition, this document defines the following terms:

Observation: Information collected directly from communications by an eavesdropper or observer. For example, the knowledge that <alice@example.com> sent a message to <bob@example.com> via SMTP taken from the headers of an observed SMTP message would be an observation.

Inference: Information extracted from analysis of information collected directly from communications by an eavesdropper or observer. For example, the knowledge that a given web page was

accessed by a given IP address, by comparing the size in octets of measured network flow records to fingerprints derived from known sizes of linked resources on the web servers involved would be an inference.

3. The Pervasive Passive Adversary

The pervasive passive adversary (PPA) is an indiscriminate eavesdropper on a computer network that can:

- observe every packet of all communications at any or every hop in any network path between an initiator and a recipient; and can
- observe data at rest in intermediate systems between the endpoints controlled by the initiator and recipient; but
- takes no other action with respect to these communications (i.e., blocking, modification, injection, etc.).

We note that a threat model that limits the adversary to being completely passive may under-represent the threat to communications privacy posed especially by well-resourced adversaries, but submit that it represents the maximum capability of a single entity interested in remaining undetectable.

The techniques available to the PPA are direct observation and inference. Direct observation involves taking information directly from eavesdropped communications - e.g., URLs identifying content or email addresses identifying individuals from application-layer headers. Inference, on the other hand involves analyzing eavesdropped information to derive new information from it; e.g., searching for application or behavioral fingerprints in observed traffic to derive information about the observed individual from them, in absence of directly-observed sources of the same information.

We would like to assume that the PPA does not have the ability to observe communications on trusted systems at either the initiator or a recipient of a communication, as there would seem to be little that a protocol designer could do in the case of compromised endpoints. However, given the state of vulnerability of many endpoints to various security exploits, we would encourage protocol designers to consider the protections their protocols afford to the privacy of their users even in the face of partially compromised endpoints.

The PPA may additionally have have privileged information allowing the reversal of strong encryption -- e.g. compromised key material or knowledge of weaknesses in the design or implementation of

cryptographic algorithms or random number generators at the initiator, recipient, and/or intermediaries. However, we consider the evaluation and improvement of cryptographic protections, while important to improving the security of the Internet in the face of pervasive surveillance, to be out of scope for this work: here, we will assume that a given cryptographic protection for a protocol works as advertised.

4. Threat analysis

On initial examination, the PPA would appear to be trivially impossible to defend against. If the PPA has access to every byte of every packet of a communication, then full application payload and content is available for applications which do not provide encryption.

Guidance to protocol designers [RFC3365] to provide cryptographic protection of confidentiality in their protocols improves this situation a great deal. The use of TLS [RFC5246] reduces the information available for correlation to the network and transport layer headers (e.g. source and destination IP addresses and ports) on each hop, but leaves any data at rest used by a protocol on intermediate systems vulnerable to intermediate system compromise.

End-to-end approaches (e.g. S/MIME [RFC3851]) help defend against this threat. However, protocols that route messages based on recipient identifier or pseudonym, such as SMTP [RFC2821] and XMPP [RFC6120], still require intermediate systems to handle these in the clear, and may leak additional metadata as well (e.g., in the S/MIME example, the SMTP headers), making this available to the PPA if it is has compromised these intermediate systems.

We can assume that the PPA does not have unlimited resources, i.e., that it will attempt to eavesdrop at the most efficient observation point(s) available to it, and will collect as little raw data as necessary to support its aims. This allows us to back away from this worst-case scenario. Storing full packet information for a fully-loaded 10 Gigabit Ethernet link will fill one 4TB hard disk (the largest commodity hard disk available as of this writing) in less than an hour; storing network flow data from the same link, e.g. as IPFIX Files [RFC5655], requires on the order of 1/1000 the storage (i.e., 4GB an hour). Metadata-based surveillance approaches are therefore more scalable for pervasive surveillance, so it is worthwhile to analyze information which can be inferred from various network traffic capture and analysis techniques other than full packet observation.

In the remainder of this analysis, we categorize the ways that information radiates off of protocols on the Internet. First, we list kinds of information that can be directly observed; this may seem somewhat obvious, but is included for completeness. We then explore the types of information which may be useful for drawing inferences about user behavior, then go into practical detail on inference attacks against just information available in the IP header, to better illustrate the extent of the problem.

4.1. Information subject to direct observation

Protocols which do not encrypt their payload make the entire content of the communication available to a PPA along their path. Following the advice in [RFC3365], most such protocols have a secure variant which encrypts payload for confidentiality, and these secure variants are seeing ever-wider deployment. A noteworthy exception is DNS [RFC1035], as DNSSEC [RFC4033] does not have confidentiality as a requirement. This implies that all DNS queries and answers generated by the activities of any protocol are available to a PPA.

Protocols which encrypt their payload using an application- or transport-layer encryption scheme (e.g. TLS [RFC5246]) still expose all the information in their network and transport layer headers to a PPA, including source and destination addresses and ports. IPsec ESP [RFC4303] further encrypts the transport-layer headers, but still leaves IP address information unencrypted; in tunnel mode, these addresses correspond to the tunnel endpoints. Cryptographic protocols themselves, e.g. the TLS session identifier, may leak information that can be used for correlation and inference. While this information is much less semantically rich than the application payload, it can still be useful for the inferring an individual's activities.

Protocols which imply the storage of some data at rest in intermediaries leave this data subject to observation at a PPA that has compromised these intermediaries, unless the data is encrypted end-to-end by the application layer protocol, or the implementation uses an encrypted store for this data.

4.2. Information useful for inference

Inference is information extracted from later analysis of an observed communication, and/or correlation of observed information with information available from other sources. Indeed, most useful inference performed by a PPA falls under the rubric of correlation. The simplest example of this is the observation of DNS queries and answers from and to a source and correlating those with IP addresses with which that source communicates can give access to information

otherwise not available from encrypted application payloads (e.g., the Host: HTTP/1.1 request header when HTTP is used with TLS).

Inference can also leverage information obtained from sources other than direct traffic observation. Geolocation databases, for example, have been developed map IP addresses to a location, in order to provide location-aware services such as targeted advertising. This location information is often of sufficient resolution that it can be used to draw further inferences toward identifying or profiling an individual.

Social media provide another source of more or less publicly accessible information. This information can be extremely semantically rich, including information about an individual's location, associations with other individuals and groups, and activities. Further, this information is generally contributed and curated voluntarily by the individuals themselves: it represents information which the individuals are not necessarily interested in protecting for privascy reasons. However, correlation of this social networking data with information available from direct observation of network traffic allows the creation of a much richer picture of an individual's activities than either alone. We note with some alarm that there is little that can be done from the protocol design side to limit such correlation by a PPA, and that the existence of such data sources in many cases greatly complicates the problem of protecting privacy by hardening protocols alone.

4.3. On the Non-Anonymity of IP Addresses

In this section, we explore the non-anonymity of even encrypted IP traffic by examining some inference techniques for associating a set of addresses with an individual, in order to illustrate the difficulty of defending communications against a PPA. Here, the basic problem is that information radiated even from protocols which have no obvious connection with personal data can be correlated with other information which can paint a very rich behavioral picture, that only takes one unprotected link in the chain to associate with an identity.

4.3.1. Analysis of IP headers

Internet traffic can be monitored by tapping Internet links, or by installing monitoring tools in Internet routers. Of course, a single link or a single router only provides access to a fraction of the global Internet traffic. However, monitoring a number of high capacity links or a set of routers placed at strategic locations provides access to a good sampling of Internet traffic.

Tools like IPFIX [RFC7011] allow administrators to acquire statistics about sequences of packets with some common properties that pass through a network device. The most common set of properties used in flow measurement is the "five-tuple" of source and destination addresses, protocol type, and source and destination ports. These statistics are commonly used for network engineering, but could certainly be used for other purposes.

Let's assume for a moment that IP addresses can be correlated to specific services or specific users. Analysis of the sequences of packets will quickly reveal which users use what services, and also which users engage in peer-to-peer connections with other users. Analysis of traffic variations over time can be used to detect increased activity by particular users, or in the case of peer-to-peer connections increased activity within groups of users.

4.3.2. Correlation of IP addresses to user identities

In Section 4.3.1, we have assumed that IP addresses can be correlated with specific user identities. This can be done in various ways.

Tools like reverse DNS lookup can be used to retrieve the DNS names of servers. Since the addresses of servers tend to be quite stable and since servers are relatively less numerous than users, a PPA could easily maintain its own copy of the DNS for well-known or popular servers, to accelerate such lookups.

On the other hand, the reverse lookup of IP addresses of users is generally less informative. For example, a lookup of the address currently used by one author's home network returns a name of the form "c-192-000-002-033.hsd1.wa.comcast.net". This particular type of reverse DNS lookup generally reveals only coarse-grained location or provider information.

In many jurisdictions, Internet Service Providers (ISPs) are required to provide identification on a case by case basis of the "owner" of a specific IP address for law enforcement purposes. This is a reasonably expedient process for targeted investigations, but pervasive surveillance requires something more efficient. A PPA that could secure the cooperation of the ISP could correlate IP addresses and user identities automatically.

Even if the ISP does not cooperate, identity can often be obtained via inference. We will discuss in the next section how SMTP and HTTP can leak information that links the IP address to the identity of the user.

4.3.3. Monitoring messaging clients for IP address correlation

POP3 [RFC1939] and IMAP [RFC3501] are used to retrieve mail from mail servers, while a variant of SMTP [RFC5321] is used to submit messages through mail servers. IMAP connections originate from the client, and typically start with an authentication exchange in which the client proves its identity by answering a password challenge.

If the protocol is executed in clear text, monitoring services can "tap" the links to the mail server, retrieve the user name provided by the client, and associate it with the IP address used to establish the connection.

The same attack can be executed against the SIP [RFC3261] protocol, if the connection between the SIP UA and the SIP server operates in clear text

In addition, there are many instant messaging services operating over the Internet using proprietary protocols. If any of these proprietary protocols includes clear-text transmission of the user identity, these can be observed to provide an association between the user identity and the IP address.

4.3.4. Retrieving IP addresses from mail headers

SMTP [RFC5321] requires that each successive SMTP relay adds a "Received" header to the mail headers. The purpose of these headers is to enable audit of mail transmission, and perhaps to distinguish between regular mail and spam. Here is an extract from the headers of a message recently received from the "perpass" mailing list:

```
 Received: from 192-000-002-044.zone13.example.org (HELO ?192.168.1.100?)
(xxx.xxx.xxx.xxx)
by lvps192-000-002-219.example.net with ESMTPSA
(DHE-RSA-AES256-SHA encrypted, authenticated);
27 Oct 2013 21:47:14 +0100
 Message-ID: <526D7BD2.7070908@example.org>
 Date: Sun, 27 Oct 2013 20:47:14 +0000
 From: Some One <some.one@example.org>
```

This is the first "Received" header attached to the message by the first SMTP relay. For privacy reasons, the field values have been anonymized. We learn here that the message was submitted by "Some One" on October 27, from a host behind a NAT (192.168.1.100) [RFC1918] that used the IP address 192.0.2.44. The information remained in the message, and is accessible by all recipients of the "perpass" mailing list, or indeed by any PPA that sees at least one copy of the message.

A PPA that can observe sufficient email traffic can regularly update the mapping between public IP addresses and individual email identities. Even if the SMTP traffic was encrypted on submission and relaying, the PPA can still receive a copy of public mailing lists like "perpass".

Similar information is available in the SIP headers [RFC3261].

4.3.5. Tracking address use with web cookies

Many web sites only encrypt a small fraction of their transactions. A popular pattern was to use HTTPS for the login information, and then use a "cookie" to associate following clear-text transactions with the user's identity. Cookies are also used by various advertisement services to quickly identify the users and serve them with "personalized" advertisements. Such cookies are particularly useful if the advertisement services want to keep tracking the user across multiple sessions that may use different IP addresses.

As cookies are sent in clear text, a PPA can build a database that associates cookies to IP addresses for non-HTTPS traffic. If the IP address is already identified, the cookie can be linked to the user identify. After that, if the same cookie appears on a new IP address, the new IP address can be immediately associated with the pre-determined identity.

4.3.6. Tracking address use with network graphs

A PPA can track traffic from an IP address not yet associated with an individual to various public services (e.g. websites, mail servers, game servers), and exploit patterns in the observed traffic to correlate this address with other addresses that show similar patterns. For example, any two addresses that show connections to the same IMAP or webmail services, the same set of favorite websites, and game servers at similar times of day may be associated with the same individual. Correlated addresses can then be tied to an individual through one of the techniques above, walking the "network graph" to expand the set of attributable traffic.

5. Evaluating protocols for PPA resistance

Though inference by a PPA makes the problem of guaranteeing privacy in the face of passive surveillance difficult, it is possible to strengthen each link in the chain in order to increase their resistance. PPA resistent protocols have the following properties:

- The confidentiality of all information not absolutely required for the operation of the protocol at intermediate systems is cryptographically protected.
- The confidentiality of all identifiers which can be associated with specific individuals through observation or inference are cryptographically protected on a hop-by-hop basis, even if they are required for the operation of the protocol at intermediate systems.
- Identifiers required for the operation of the protocol are non-persistent and non-specific to individuals to the extent possible.
- The protocol radiates as little information as possible which can be used to fingerprint specific instances of the protocol.

Clearly, the messaging protocols examined in Section 4.3 are, by these criteria, not particularly resistent to a PPA. In evaluating a protocol for PPA resistance, tradeoffs in efficiency, latency, manageability, and other application requirements will need to be evaluated, as well. More detailed information on privacy considerations for protocol design are given in [I-D.cooper-ietf-privacy-requirements].

6. General protocol design recommendations for PPA resistance

The following general recommendations are intended to guide discussions about improving the resistance of IETF protocols to a PPA; specific recommendations are the subject of a separate specification.

6.1. Encrypt everything you can

Though IETF protocols have been long moving in the direction of more and better cryptographic protection [RFC3365], there is continued room for improvement. Approaches such as opportunistic encryption, while not providing identity guarantees, may have benefits in confidentiality that reduce the information radiated from protocols, increasing the costs for pervasive surveillance. To some extent encryption is a deployment problem rather than a protocol design and implementation problem; improvements in usability may be useful here.

The design and deployment of end-to-end encryption for a protocol, especially for messaging applications, can reduce the ability of a PPA to observe application-layer information and identifiers at a compromised intermediate system.

6.2. Design and implement for simplicity and auditability

This would seem to be common sense, but in practice it is not really the case that protocol design processes naturally have simplicity as a goal. Simplicity of a design is directly related to the auditability of the design and implementations thereof. Privacy and security features designed into a protocol which are too complex to understand will suffer from limited implementation and deployment. A good example of such a case is IPsec where primary complaints are related to its complexity [Ferguson03].

The auditability of a protocol is directly related to the ability to measure and reason about the information that it radiates that could be used for inference by a PPA. Audits of designs and implementations can also reduce the risk of hidden side channels which could carry additional information useful to a PPA. One approach for improving auditability is the release of implementations as open source.

6.3. Allow for fingerprinting resistance in protocol designs

Fingerprinting provides a source of information for inference, and can rely on packet and flow size and timing information. The inclusion of null information in packets, or grouping information into more/fewer packets can reduce this risk. Since protocols tend to be optimized for minimum bandwidth usage and minumum latency, the only way to go is up, so this resistance comes at the expense of usable bandwidth and increased latency. While not necessarily applicable in the general case, protocol designs can make it possible do to this.

6.4. Do not rely on static IP addresses

Always on broadband connections may or may not provide the subscribers with static IP addresses. Some users pay extra for the convenience of a stable address. Of course, stable addresses greatly facilitate IP header monitoring.

In contrast, we could imagine that the broadband modem is re-provisioned at regular interval with a new IPv4 address, or with a new IPv6 address prefix. Some convenience will be lost, and TCP connections active before the renumbering will have to be reestablished. However, the renumbering will significantly complicate the task of IP header monitoring.

Similarly, the Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [RFC4941] allow users to configure temporary IPv6 addresses out of a global prefix. Privacy addresses are meant to be used for a short time, typically no more than a day, and are specifically designed to render monitoring based on IPv6 addresses harder.

7. IANA Considerations

This document has no actions for IANA

8. Security Considerations

This document explores the capabilities of an adversary with an interest in undermining the security of the Internet to enable pervasive surveillance activities. It does not provide any specific protocol guidance that may impact the security of those protocols, but it is hoped that the awareness of this threat will end up being a metacontribution to Internet security.

9. Acknowledgments

Thanks to Dilip Many and Stephan Neuhaus, who contributed to an initial version of this work. Thanks to Mark Townsley, Stephen Farrell, Chris Inacio, and others in the anonymity set of "people we've forgotten to thank" for feedback and input to this draft.

10. References

10.1. Normative References

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[I-D.cooper-ietf-privacy-requirements] Cooper, A., Farrell, S., and S. Turner, "Privacy Requirements for IETF Protocols", draft-cooper-ietf-privacy-requirements-01 (work in progress), October 2013.

10.2. Informative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, August 2002.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

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

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

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

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5655] Trammell, B., Boschi, E., Mark, L., Zseby, T., and A. Wagner, "Specification of the IP Flow Information Export (IPFIX) File Format", RFC 5655, October 2009.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, September 2013.

[Ferguson03] Ferguson, D. and B. Schneier, "A Cryptographic Evaluation of IPsec (https://www.schneier.com/paper-ipsec.pdf)", December 2003.

Authors' Addresses

Brian Trammell
Swiss Federal Institute of Technology Zurich
Gloriastrasse 35
8092 Zurich
Switzerland

Phone: +41 44 632 70 13
Email: trammell@tik.ee.ethz.ch

Daniel Borkmann
Red Hat
Seefeldstrasse 69
8008 Zurich
Switzerland

Email: dborkman@redhat.com

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
U.S.A.

Email: huitema@huitema.net