Pretty Good Packet Authentication

Andreas Haeberlen[†‡] Rodrigo Rodrigues[†] Krishna Gummadi[†] Peter Druschel[†]

[†]*Max Planck Institute for Software Systems (MPI-SWS)* [‡]*Rice University*

Abstract

Internet addresses are routinely being used to infer the identity of persons who send offending traffic – a capability they were not designed to provide. As a result, problems abound: innocent users are being accused, while the culprits can easily avoid detection.

In this paper, we present Pretty Good Packet Authentication (PGPA), a simple service that can establish whether or not a given host has sent a particular packet. PGPA provides a firm basis on which to act against the culprit, and, at the same time, it enables innocent users to defend themselves against false accusations. We also describe an implementation of PGPA that can be deployed incrementally and with minimal changes to the current Internet.

1 Introduction

The Internet lacks a mechanism to reliably verify the source of a delivered packet. Nevertheless, the source IP address in a packet is routinely used to determine the identity of the end node responsible for sending the packet. ISPs, spam filters, and firewalls use IP addresses to black-list and block traffic from compromised or malicious end nodes [8]. Furthermore, law enforcement agencies have begun to use packet source IP addresses as evidence when prosecuting users who allegedly distribute protected or illegal content over the Internet [5, 7]. However, recent incidents show that it is risky to rely on IP addresses to determine the source of Internet traffic.

On the one hand, because IP addresses in the Internet can be easily spoofed, malicious users can use forged addresses to evade detection. In fact, it is well known that vulnerabilities in the interdomain routing protocol, BGP, can be exploited to hijack entire IP prefixes for short periods of time. The hijacker can send and receive traffic from the hijacked pool of IP addresses during this time. For example, a recent study of email spam has shown that some spammers use this technique to work around black-listed IP addresses [8]. Worse, under the current state of affairs, the victim could be falsely accused and held responsible for actions of the attacker.

Recently, researchers at the University of Washington demonstrated the potential for such false accusations by causing hundreds of DMCA take-down notices to be generated for several machines that were not participating in file sharing networks, including their own printer and a network access point [7]. In a more serious case, an innocent user was suspected of having downloaded child pornography because his ISP had made a mistake when looking up an IP address for the police. By the time the mistake was discovered, the police had already confiscated his computers and searched his home [5].

On the other hand, if an offending packet *does* have the correct source address, the sender can use the weakness of the current mechanism as an excuse. Since one cannot be sure whether the packet's source IP address was spoofed or not, the real sender can plausibly deny having sent the packet. Thus, it is difficult to hold users accountable for their actions.

In this paper, we present a system called *Pretty Good Packet Authentication (PGPA)*, which addresses these problems by offering a simple packet authentication service. PGPA determines whether or not a host H has sent a specific packet P at approximately time t. This provides a firm basis on which to act against the owners of hosts that send malicious or illegal traffic, since they can no longer deny having sent it, or use the lack of reliable authentication as an excuse. At the same time, PGPA can establish that a suspected device has *not* sent a particular packet, which gives innocent users the ability to defend themselves against false accusations.

Unlike systems that are designed for online verification of the traffic source [1], PGPA authenticates packets not just to their recipients, but to anyone who has access to them. For example, a system administrator can use intercepted packets to convince the police that

a certain user has attempted to break into one of his systems, and the police can query PGPA to verify his claims. With an online verification system, an intercepted packet would not be convincing evidence because the accused user could claim that the source address had been forged by the administrator.

PGPA also protects users' privacy. Since PGPA only answers queries about specific packets and specific points in time, it cannot be used to spy on the user's traffic or to screen user's traffic for suspicious behavior. In order to get useful information out of PGPA, the inquirer must either be the recipient of the packets in question, or he must have intercepted them or obtained them from the recipient. In all cases, the content of the packets and the suspected source is already known to the inquirer; PGPA merely affirms the authenticity of the source address.

On the one hand, PGPA is a simple service; it does not attempt to match the functionality of full-blown accountability systems like AIP [1]. On the other hand, it has the advantage of being much easier to deploy than a clean-slate solution. We show that PGPA can be implemented on the user's Internet access link, without requiring changes to routers or to the Internet backbone, and that it can be incrementally deployed.

The rest of this paper is structured as follows. We give a brief overview and discuss related work in Section 2, we outline an implementation of PGPA in Section 3, and we discuss several applications in Section 4. In Section 5, we conclude the paper and briefly outline future work.

2 Overview and related work

In this section, we give a brief overview of packet authentication, and we discuss related work.

2.1 Packet authentication

The strongest form of packet authentication (in which anyone can independently verify the authenticity of any packet at any time) would be difficult to achieve. A strawman solution would require that each packet be cryptographically signed, as well as a public-key infrastructure that allows other hosts to verify these signatures. This solution would enable the use of packets as evidence, since anyone could verify the signatures and thus link a packet to its sender. However, such a solution would be computationally expensive and difficult to deploy.

We propose a weaker form of packet authentication, which enables an ISP to verify whether a given, recently transmitted IP packet was sent by one of its customers. Stated more precisely, PGPA provides the following capability:

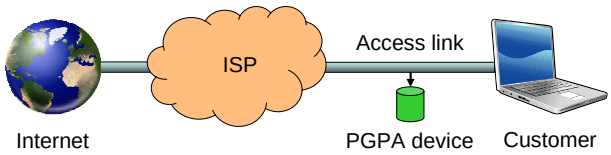

Figure 1: Example PGPA setup. The PGPA device keeps a digest of the packets the customer transmits. Given a packet and the approximate time of transmission, it can verify whether the packet is authentic.

- Given a packet P with source IP address S and a timestamp t, the ISP that owns S can verify whether S has sent P at approximately time t.

We will show that this capability can be implemented and deployed in a straightforward manner.

2.2 Prior work

Previous systems have provided approximate solutions for this problem. In particular, IP traceback systems [9, 10] can identify the source of network traffic; however, they require routers to mark packets and/or store information about them. This would involve major changes to current networks, which we are trying to avoid.

On another related topic, accountability on the Internet has been proposed as the ability to associate an action with the responsible entity. The Accountable Internet Protocol (AIP) [1] is a replacement for IP that allows hosts and domains to prove that they own the address they use. Based on this foundation, the protocol aims at dealing with source spoofing, route spoofing and denial-of-service (DoS). However, AIP is a clean-slate design whose deployment would require upgrading major components of the Internet, including the network protocol and the addressing structure.

In this paper, we settle for a narrower goal: verifying that a packet was sent from the host whose IP address appears in the packet's header. We show that this simple capability can be deployed quite easily and still provides powerful properties.

2.3 How would a PGPA system work?

Packet authentication essentially requires setting up a device that keeps track of transmitted packets. Where in the network should this device be placed? A natural choice is the customer's access link, that is, the link between the customer and his ISP (Figure 1). Here, the real identity of the customer is known from his contract with the ISP, and the device can observe the customer's entire traffic.

Deploying packet authentication on the access links has three distinct advantages. First, it does not require upgrades or additional processing in the high-speed

backbone. Second, it helps against customer-side spoofing because the ISP knows which IP addresses have been assigned to each link (e.g., via DHCP) and can filter out any packets with different addresses. Finally, it provides for incremental deployment, since each access link can be upgraded independently. This provides us with a feasible deployment path.

2.4 Is PGPA sufficient?

PGPA can tie a packet to an access link, and via the ISP's records, to the customer attached to that link. PGPA cannot by itself identify the person who is responsible for sending a packet. The latter would require, in addition to PGPA, that each device run a secure operating system that authenticates users and cannot be compromised. However, PGPA is still a significant improvement over the state of the art because it can tie any packet to the owner of the Internet account from where the packet was sent. Ensuring that only authorized users can send packets from a given account or device is a complementary problem and beyond the scope of PGPA.

2.5 Who can we trust?

We assume that ISPs that participate in PGPA do not collude with customers. This assumption is necessary because an ISP could allow a customer to send packets in a way that circumvents PGPA, making it appear as if the packet had a spoofed IP address. We believe it is a reasonable assumption that reputable ISPs do not actively protect malicious customers. If some ISPs are known not to be trustworthy, this must be taken into account when judging their PGPA responses.

A different question is whether an ISP must be trusted by its customers not to blame them (accidentally or deliberately) for packets they did not actually send. This depends on how PGPA is implemented; we will discuss the details in Section 3.2.

3 Implementing PGPA

In this section, we describe a simple system that implements PGPA in the current Internet architecture.

3.1 The traffic monitor

PGPA requires a new type of middlebox called a *traffic monitor*, which records information about packets that have been transmitted over a given access link. For each packet, the traffic monitor calculates a digest and a timestamp, and it writes them to a built-in storage device. When the user is accused of having sent illegal or offensive traffic, the accuser must produce at least one packet as evidence, together with the approximate time of its transmission. The ISP then queries the traffic monitor for a digest that matches the evidence. If such a digest is found and its timestamp is close enough, the user is held responsible for the traffic.

3.2 Location of the monitor

The traffic monitor can be installed at either end of the access link, that is, either on the user's premises or at the ISP. Installing the monitor on the user's premises may be more scalable and less expensive for the ISP. In addition, the user does not have to trust the ISP, because he can install a monitor device from a vendor he trusts. On the other hand, if the monitor is under the user's control, the ISP must prevent users from bypassing the monitoring device. The ISP could achieve this by establishing a secure channel to the monitor and only accepting traffic via this channel. For example, a secret key could be shared between the monitor and the ISP, and the device could append a MAC to each packet it sends to the ISP. To guard against tampering, the device could erase the key material when it detects that the case has been opened.

Of course, if the device is physically within the user's reach, he can destroy the device, and thus the evidence it contains. Hence, user-side monitors cannot be used to fight severe crime. Even though the destruction of the device would raise suspicion and might even be punishable by itself, that punishment may be preferable to the penalty for the crime. If this is a concern, the monitors should be deployed at the ISP instead.

3.3 Calculating digests

The monitor stores digests rather than full packets. Digests are sufficient to check for the presence or absence of a specific packet, and they do not reveal the contents of other transmitted packets. However, we must be careful when computing the digest because in IP networks, certain header fields of a packet can change along the path, such as the TTL and the checksum. To prevent false negatives because the hash of the evidence packet does not match the recorded hash, the monitor masks out these fields before computing the hash, e.g., using the technique presented in [10].

Packets can also be transformed by IP fragmentation. A simple way to handle fragmentation is to have the monitor reassemble outgoing packets before calculating their digests, and to allow PGPA queries only for complete packets. If an observer intercepts packet fragments, he must reassemble the corresponding packets and issue a PGPA query for the reassembled packet.

3.4 Storage requirements

With finite storage, the monitor can only keep its records for a limited amount of time. This is sufficient because the hashes only need to be stored until the accusation is first made; old hashes can be deleted. If long-term authentication is required, the monitor can answer queries with a signed certificate, which then remains valid even when the original records are no longer available.

An inexpensive hard drive should be sufficient for almost all access links – especially since most connections are asymmetric and have a low-capacity upstream. Today, top-of-the-line DSL connections have a capacity of 1 Mbps, so they can transmit at most 3,125 40-byte packets[1] per second. Hence, even under worst-case assumptions, a 187 GB hard drive is sufficient to keep a SHA-1 hash and a 32-bit timestamp for a month. Furthermore, since storage capacity has grown faster than bandwidth [3], future technology will tend to make it possible to store the necessary data for increasingly longer periods. Having storage at the access link is increasingly common; service providers are already deploying managed boxes, such as TriplePlay gateways or set-top boxes [6], many of which already include a storage device with substantial capacity.

3.5 Preventing information leaks

A potential concern with PGPA is that it might inappropriately leak information about users' traffic. For example, suppose an attacker wanted to know whether the user had accessed a certain web site. Could the attacker simply guess a few packets that the user would have sent to the site if he had accessed it (such as TCP acknowledgment packets on port 80) and then query PGPA to see whether any of these packets had actually been sent?

The chances of success for such an attack are low, because the attacker would have to guess not only the entire contents of a packet – including 'random' fields such as the TCP sequence number, the TCP acknowledgment number and the IP identifier – but also the approximate time of its transmission. PGPA must allow a small time window when checking timestamps; after all, there are queueing delays and clocks on the Internet are only loosely synchronized, so we cannot expect the requester to know the exact time when the packet was sent. However, this window can be kept small, on the order of seconds. It would be difficult for the attacker to correctly guess all of the above header fields (80 bits) plus the approximate time of transmission. Furthermore, the monitor can introduce additional randomness by replacing the IP identifier, or by adding a header option with a random value. Finally, the rate at which the monitor answers queries can be limited to prevent search attacks.

If the monitor device is stolen and compromised, the attacker could run a dictionary attack on the stored hash values. To mitigate the risk of information leakage in this case, the monitor can include a salt value when calculating the hashes.

[1] This corresponds to a TCP acknowledgment; data packets are usually much larger.

3.6 Guarantees

Compared to our initial definition of PGPA in Section 2.1, the traffic monitor introduces two practical limitations. First, due to its finite storage capacity, the monitor can only provide information about packets that are not older than a certain amount of time T_{max}, say one month. This should be acceptable in practice as long as misbehavior can typically be detected within that time. Second, the system can only answer queries when a monitor has been deployed on the alleged sender's access link.

Thus, the guarantee we achieve is: Given a packet P, a timestamp t *that is not more than* T_{max} *in the past*, and a source IP address S, the ISP that owns S can verify whether S has sent P at approximately time t, *provided that monitors are deployed on the ISP's access links*.

4 Applications of PGPA

Next, we discuss applications and use cases for the proposed IP packet authentication facility.

The facility allows an ISP to verify whether a given packet that was allegedly sent by one of its customers is authentic. This capability is useful whenever one of the ISP's customers is accused of having sent offending traffic.

For instance, ISPs frequently receive complaints from copyright holders or law enforcement agencies, alleging that one of the ISP's customers has participated in illegal activity on the Internet and demanding that the ISP reveal the identity of the customer associated with a given IP address. Today, ISPs have to take such demands at face value and reveal the customer's identity. When the allegation turns out to be incorrect, this may be tragic for the customer and embarrassing for the ISP. For instance, among many recent cases, an IP address that was implicated as having downloaded copyrighted material from a BitTorrent turned out to be that of a printer [7]. Using a packet authentication facility, an ISP can check if an allegation is well-founded. It can demand to be presented a packet trace of the offending traffic and check if matching hash values are found in the associated customer's traffic digest.

PGPA can also be used to improve existing schemes to mitigate denial of service attacks. For instance, a

"shut-up" service was proposed, which allows a host D to ask that a particular host S who is sending traffic to D be prevented from sending any further traffic [4]. Packet authentication can be used to validate such requests, thereby preventing abuse of the "shut-up" service.

From the perspective of a customer, IP packet authentication provides a defense against false accusations. We have already mentioned the case of [5], where an innocent user was accused of having downloaded child pornography. Packet authentication would have allowed the user's ISP to establish that the offending traffic was not sent by its customer, and thus to protect him from false accusation and slander. Even in the case of a clerical error, such as a mixed-up IP address, the user could easily have proved his innocence by agreeing to an inspection of his monitor device.

Finally, PGPA can serve as a building block for a global IP packet authentication service, which enables anyone on the Internet to verify the authenticity of any IP packet they receive. The service would enable hosts to present an arbitrary IP packet and ask if the packet is authentic. The service would determine the ISP (AS) responsible for the packet's source IP address, and query that ISP's PGPA system. The service answers with an indication of either "authentic" (i.e., a matching packet digest is found in the source's traffic digest), "not authentic" (i.e., a matching packet digest is not found), or "unknown" (i.e., the source AS does not provide packet authentication or the packet has expired from the source's traffic digest). The service itself can be secured from manipulation using techniques similar to those used in DNSSEC [2].

5 Conclusion and future work

In this paper, we have presented PGPA, a simple service that allows an ISP to determine whether one of its customers has sent a given packet at some point in the past. PGPA enables a change in the way that source IP addresses are used: today, they are an unreliable source of information that can be falsified by attackers and can cast suspicion on innocent users; with PGPA, source addresses could become a key element of forensic analysis, enabling a reliable confirmation of the traffic source, and allowing victims of false accusations to prove their innocence. PGPA is simple to implement, works in partial deployments, and has a plausible deployment path. We have also shown several applications that can benefit from using PGPA. We are planning to prototype the traffic monitor by modifying a wireless access point, and to implement a packet authentication service.

However, PGPA is only a first step towards more accountability on the Internet. Many issues cannot be addressed by PGPA alone. For example, consider a situation where a user's access link is shared, possibly without his knowledge or consent (perhaps through an open wireless access point). If someone who is sharing the user's access link misbehaves, the user cannot prove his innocence using PGPA. A similar situation arises when the user's machine sends illicit traffic without the user's knowledge, for example, if the machine has been infected by malware. We leave solutions for these as future work.

References

[1] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable internet protocol (AIP). In *Proceedings of SIGCOMM*, Aug 2008.

[2] D. E. Eastlake. RFC 2535: Domain name system security extensions. `http://www.faqs.org/rfcs/rfc2535.html`, March 1999.

[3] J. Gray. Distributed computing economics. Technical Report MSR-TR-2003-24, Microsoft Research, Mar 2003.

[4] S. Guha, P. Francis, and N. Taft. ShutUp: End-to-end containment of unwanted traffic. Technical Report `http://hdl.handle.net/1813/11101`, July 2008.

[5] Heise.de. IP-Verwechslung führt zu falschem Kinderporno-Verdacht. `http://www.heise.de/newsticker/meldung/105094`, 2008.

[6] N. Laoutaris, P. Rodriguez, and L. Massoulie. ECHOS: Edge capacity hosting overlays of nano data centers. *SIGCOMM Comput. Commun. Rev.*, 38(1):51–54, 2008.

[7] M. Piatek, T. Kohno, and A. Krishnamurthy. Challenges and directions for monitoring P2P file sharing networks. In *3rd USENIX Workshop on Hot Topics in Security (HotSec '08)*, July 2008.

[8] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proceedings of SIGCOMM'06*, Sept. 2006.

[9] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of SIGCOMM'00*, Aug 2000.

[10] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. Single-packet IP traceback. *IEEE/ACM Trans. Netw.*, 10(6):721–734, 2002.