Digital Village

Hal Berghel and Kim Womack

Anonymizing the Net

Sanitizing packets for fun and profit.

The technique of "anonymizing" network traffic has existed for many years. In its most basic form, an 'anonymizer' is the combination of software and some network appliance (server, router, gateway) that redirects network traffic in such a way that the primary functionality is preserved while all identifying characteristics of the traffic that might enable a network analyst to trace the traffic back to the original source are removed. In practice, anonymizing is associated with both Web and email network activity, although in the latter case the term "remailer" is usually used to describe the process. However, in principle any network traffic, and any network protocol, can be anonymized.

The general practice of anonymization can be easily described by reference to a typical IP packet. Figures 1 and 2 reveal the contents of two TCP/IP packets from the point of view of an Internet client. Figure 1 is the outbound packet to the server (or 'anonymizer'), and Figure 2 is the inbound packet from the server (or 'anonymizer').[1] The connection between the SYN in Figure 1 and the ACK in Figure 2 is evident from the sequence numbers.

A very brief analysis of some of these fields should prove helpful to the discussions that will follow.

Version number field. Currently, most Internet traffic is Version 4 as in the examples shown in the figures here. Its high-bandwidth offspring is Version 6 (IPv6) which is currently being deployed on Internet2.

Internet Header Length field denotes the length of the packet header in 4-byte words before the actual packet payload begins. In this case, the header length is 5x4=20 bytes, which is the typical IP header length. The payload of the IP header in this case is the TCP packet of 28 (48-20) bytes.

The *Time to Live* field contains the number of hops left before the packet expires, in this case 128 in the outbound packet from a Windows 2000 client, and 64 in the inbound packet from the Linux server/ anonymizer. The idea is to terminate lost or errant traffic by decrementing the TTL by one every time a packet passes through a network router or gateway. Different operating systems have different initial TTLs, so this can be an identifying piece of data.

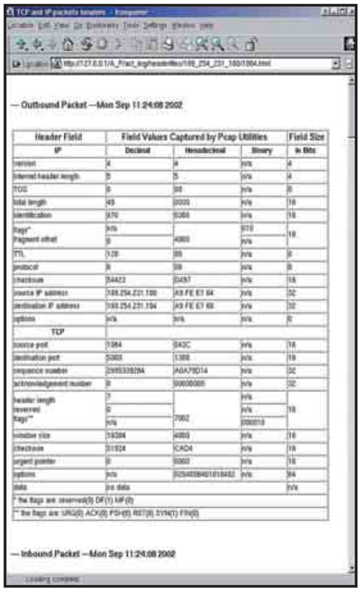
Protocol field designates the IP

[1] Note that we have put the term 'anonymizer' in scare quotes throughout this column. According to Lance Cottrell, CEO of Anonymizer.com, the word is actually trademarked by Anonymizer.com, despite the fact it was a common term used by the networking community prior to Anonymizer.com's existence. To us, this is like trademarking "milk," but then we're not trademark attorneys. In deference to Mr. Cottrell, we put scare quotes around the term here whenever it is not associated with his company.

ARTUR E. GIRON

Were it to be used to prevent the target server from depositing cookies on our hard drive, we may argue that anonymization is a social good. Were the very same service to be used to foster the formation of hate groups, download pornography to minors, or support terrorism, we would likely have the opposite reaction.

protocol that defines the contents of this packet's payload. 1 is ICMP, 2 is IGMP, 6 is TCP, 17 is UDP, and so forth, for the scores of available protocols. In this case, the protocol is 6 (TCP).


TCP and IP packets headers - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location http://127.0.0.1/A_Fract_tcp/headerfiles/169_254_231_100/1084.html

— Outbound Packet —Mon Sep 11:24:08 2002

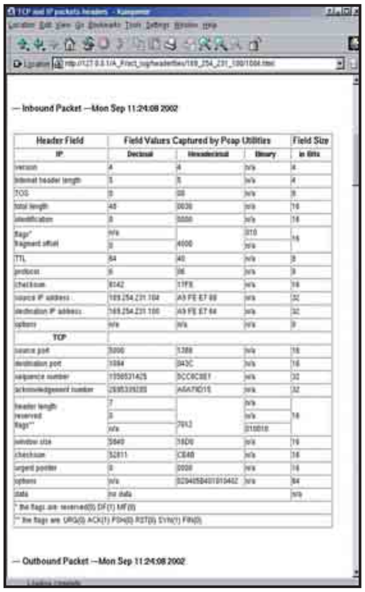
Header Field	Field Values Captured by Pcap Utilities			Field Size
IP	Decimal	Hexadecimal	Binary	in Bits
version	4	4	n/a	4
internet header length	5	5	n/a	4
TOS	0	00	n/a	8
total length	48	0030	n/a	16
identification	876	036C	n/a	16
flags*	n/a		010	16
fragment offset	0	4000	n/a	
TTL	128	80	n/a	8
protocol	6	06	n/a	8
checksum	54423	D497	n/a	16
source IP address	169.254.231.100	A9 FE E7 64	n/a	32
destination IP address	169.254.231.104	A9 FE E7 68	n/a	32
options	n/a	n/a	n/a	0
TCP				
source port	1084	043C	n/a	16
destination port	5000	1388	n/a	16
sequence number	2795339284	A6A79D14	n/a	32
acknowledgement number	0	00000000	n/a	32
header length	7		n/a	16
reserved	0		n/a	
flags**	n/a	7002	000010	
window size	16384	4000	n/a	16
checksum	31924	CAD4	n/a	16
urgent pointer	0	0000	n/a	16
options	n/a	020405B401010402	n/a	64
data	no data			n/a
* the flags are: reserved(0) DF(1) MF(0)				
** the flags are: URG(0) ACK(0) PSH(0) RST(0) SYN(1) FIN(0)				

— Inbound Packet —Mon Sep 11:24:08 2002

Figure 1. Outbound TCP/IP packet from Internet client (source: Anonymator).

Source IP address field specifies the IP address of originating computer. Obviously, a dead giveaway of the source. In this case, the IP addresses are internal and non-routable, since the traffic we are analyzing is on one of our experimental networks using our own proprietary 'anonymizer' known as Anonymator.

Within the TCP packet, we have additional information that is useful in characterizing and identifying the network traffic. Such fields as source port, destination port, sequence number, the flag fields, window size, and "options" data can all be used for host and destination identification by those who analyze the network traffic and packet contents. The point to this discussion is that anonymizing Internet traffic requires modification of these fields in such a way as to simultaneously achieve the intended effect of the connectivity, and prevent anyone who might analyze this traffic from identifying the source or, in some cases, the destination. Even armed with the most simple tools, such as ARIN registry or Whois, one can quite easily associate an IP address with a domain name and a contact person. So, as far as the packet headers are concerned, anonymization means sanitization.


TCP and IP packets headers - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location http://127.0.0.1/A_Fract_tcp/headerfiles/169_254_231_100/1084.html

— Inbound Packet —Mon Sep 11:24:08 2002

Header Field	Field Values Captured by Pcap Utilities			Field Size
IP	Decimal	Hexadecimal	Binary	in Bits
version	4	4	n/a	4
internet header length	5	5	n/a	4
TOS	0	00	n/a	8
total length	48	0030	n/a	16
identification	0	0000	n/a	16
flags*	n/a		010	16
fragment offset	0	4000	n/a	
TTL	64	40	n/a	8
protocol	6	06	n/a	8
checksum	6142	17FE	n/a	16
source IP address	169.254.231.104	A9 FE E7 68	n/a	32
destination IP address	169.254.231.100	A9 FE E7 64	n/a	32
options	n/a	n/a	n/a	0
TCP				
source port	5000	1388	n/a	16
destination port	1084	043C	n/a	16
sequence number	1556531425	5CC8C8E1	n/a	32
acknowledgement number	2795339285	A6A79D15	n/a	32
header length	7		n/a	16
reserved	0		n/a	
flags**	n/a	7012	010010	
window size	5840	16D0	n/a	16
checksum	52811	CE4B	n/a	16
urgent pointer	0	0000	n/a	16
options	n/a	020405B401010402	n/a	64
data	no data			n/a
* the flags are: reserved(0) DF(1) MF(0)				
** the flags are: URG(0) ACK(1) PSH(0) RST(0) SYN(1) FIN(0)				

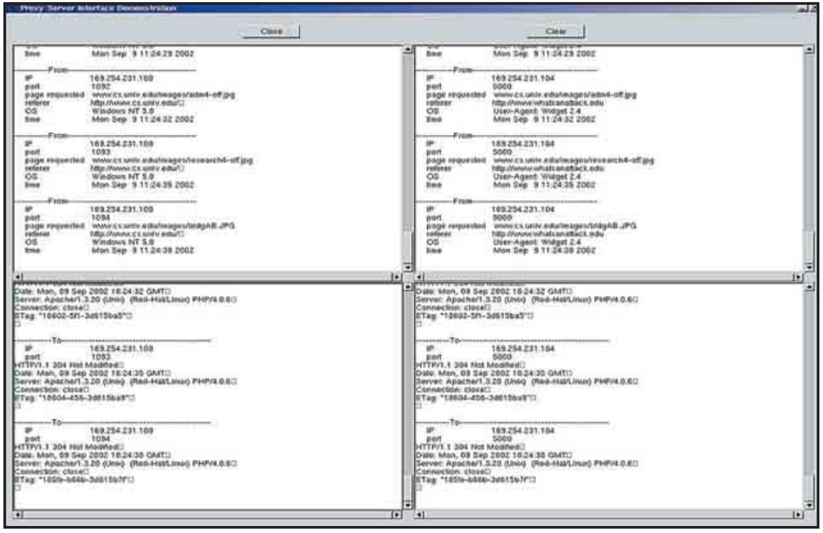
— Outbound Packet —Mon Sep 11:24:08 2002

Figure 2. Inbound TCP/IP packet to Internet client (source: Anonymator).

Figure 3 illustrates the process. In this case, the 'anonymizer', Anonymator, is of our own design.[2] The screen shot in Figure 3 is read as follows: The top row is a packet summary of the traffic as it enters (left) and exits (right) Anonymator to enter the Internet cloud. The bottom row is a packet summary of the returning traffic from the Internet as it enters (right) and leaves (left) the anonymizer to return to the client. Visual inspection confirms that the referrer HTTP field in

[2] Anonymator was created in our Cybermedia Research Lab as a general experimental platform to test for efficiencies and vulnerabilities in anonymizing technologies. We intend to release Anonymator as open-source software soon and will post the notice in a future column.

the outbound packet from the client was one of our university Web pages, while the OS field identifies the client OS as some variation of Windows NT, and the IP address is 169.254.231.100. However, after the Anonymator's sanitization, the outbound packet IP address has been changed to 169.254.231.104, while the active Web page became www.whatsanattack.edu, and the OS ID was changed to Widget 2.4. Although simplified, this illustrates the general principle.



```
time              Mon Sep  9 11:24:29 2002

-------From-----------------------------------
  IP               169.254.231.100
  port             1092
  page requested   www.cs.unlv.edu/images/adm4-off.jpg
  referer          http://www.cs.unlv.edu/□
  OS               Windows NT 5.0
  time             Mon Sep  9 11:24:32 2002

-------From-----------------------------------
  IP               169.254.231.100
  port             1093
  page requested   www.cs.unlv.edu/images/research4-off.jpg
  referer          http://www.cs.unlv.edu/□
  OS               Windows NT 5.0
  time             Mon Sep  9 11:24:35 2002

-------From-----------------------------------
  IP               169.254.231.100
  port             1094
  page requested   www.cs.unlv.edu/images/bldgAB.JPG
  referer          http://www.cs.unlv.edu/□
  OS               Windows NT 5.0
  time             Mon Sep  9 11:24:38 2002
```

```
time              Mon Sep  9 11:24:23 2002

-------From-----------------------------------
  IP               169.254.231.104
  port             5000
  page requested   www.cs.unlv.edu/images/adm4-off.jpg
  referer          http://www.whatsanattack.edu
  OS               User-Agent: Widget 2.4
  time             Mon Sep  9 11:24:32 2002

-------From-----------------------------------
  IP               169.254.231.104
  port             5000
  page requested   www.cs.unlv.edu/images/research4-off.jpg
  referer          http://www.whatsanattack.edu
  OS               User-Agent: Widget 2.4
  time             Mon Sep  9 11:24:35 2002

-------From-----------------------------------
  IP               169.254.231.104
  port             5000
  page requested   www.cs.unlv.edu/images/bldgAB.JPG
  referer          http://www.whatsanattack.edu
  OS               User-Agent: Widget 2.4
  time             Mon Sep  9 11:24:38 2002
```

```
Date: Mon, 09 Sep 2002 18:24:32 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "18602-5f1-3d615ba5"□
□

-------To-------------------------------------
  IP               169.254.231.100
  port             1093
HTTP/1.1 304 Not Modified□
Date: Mon, 09 Sep 2002 18:24:35 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "18604-456-3d615ba9"□
□

-------To-------------------------------------
  IP               169.254.231.100
  port             1094
HTTP/1.1 304 Not Modified□
Date: Mon, 09 Sep 2002 18:24:38 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "185fe-b66b-3d615b7f"□
□
```

```
Date: Mon, 09 Sep 2002 18:24:32 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "18602-5f1-3d615ba5"□
□

-------To-------------------------------------
  IP               169.254.231.104
  port             5000
HTTP/1.1 304 Not Modified□
Date: Mon, 09 Sep 2002 18:24:35 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "18604-456-3d615ba9"□
□

-------To-------------------------------------
  IP               169.254.231.104
  port             5000
HTTP/1.1 304 Not Modified□
Date: Mon, 09 Sep 2002 18:24:38 GMT□
Server: Apache/1.3.20 (Unix)  (Red-Hat/Linux) PHP/4.0.6□
Connection: close□
ETag: "185fe-b66b-3d615b7f"□
□
```

Figure 3. An illustration of packet anonymizing: A screen shot from Anonymator.

So if one should want to anonymize traffic, the starting point is to change the source IP address, modify the TCP port address, change the TTL values to inhibit OS fingerprinting, maybe "bend" the protocol to escape monitoring, and so forth. That's the starting point of what has become the practice of "Anonymizing the Internet."

Motivation

Like most everything in life, anonymization may serve both good and evil purposes. On the positive side, anonymization may ensure privacy and anonymity in support of free speech and the support of politically unpopular positions. On the negative side, it may enable such illicit or unauthorized network behavior as the downloading of pornography, using institutional email for personal use, and so forth. How might this work?

Consider the case of surfing the Web. Corporate and institutional firewalls are all capable of logging and filtering both inbound and outbound Web traffic by IP address, domain name, content, and other parameters. In this fashion, it is fairly easy to ban or censor Internet traffic. How would an 'anonymizer' change that?

The answer is that the anonymizing service becomes the man-in-the-middle. Under one design, all traffic between the client browser and the 'anonymizer' could be encrypted, thus concealing the types of identifying information discussed previously. We would presume that this outbound encrypted communication would be unreadable, even if noticed. The packet (aka communication) would be decrypted at the anonymizing server, whereupon the Web connection specified by the client would be directed to the target server. The resulting downloads would, in turn, be re-encrypted by the anonymizing server and sent along to the client. From the point of view of the client institution's firewalls, the only irregularity would be the presence of encrypted traffic, but the contents would indiscernible.

What is there to gain by anonymizing this Internet traffic? In short, the anonymization can hide a variety of information, including:

- URLs;
- Navigation history;
- The nature of downloaded content;
- What browser is being used;
- Source IP addresses; and
- Operating systems.

In short, the Web communication can be sanitized to prevent tracing the traffic beyond the 'anonymizer' back to the originating client. This capability carries with it significant societal implications. Were it to be used to prevent the target server from depositing cookies on our hard drive, we may argue that anonymization is a social good. Were the very same service to be used to foster the formation of hate groups, download pornography to minors, or support terrorism, we would likely have the opposite reaction. The point we wish to emphasize is that society should be investing much more energy thinking about this issue.

The Anonymizing Landscape

Anonymizer.com is the one of the first, if not the first, 'anonymizer' developed for the Web. Originally developed by Justin Boyan (see www.december.com/cmc/mag/1997/sep/boyan.html) in the summer of 1995 and tested at Carnegie Mellon later that year, it was subsequently licensed to C2net in 1996 then sold to Infonex in 1997.

Anonymizer.com is now one of the largest and most popular online privacy services, offering both shareware and commercial versions. Anonymizer.com's operation is faithful to the general description of the anonymizing process given earlier in this column. As packets are passed through Anonymizer.com's servers, the packet headers are sanitized of information that may identify the source of the HTTP request. In its most basic form, only the URL of the request is encrypted using the public-key encryption scheme Blowfish. In this way, the request http://www.porn.com/ would appear as outbound traffic from the originating client as http://www.anonymizer.com/ciphertext, where ciphertext would be the encrypted form of the target URL.

This approach to anonymization is mature and stable. Unfortunately, it relies heavily on trust—trust in the fact that Anonymizer.com will neither maintain nor release privileged information, and trust that third-party attacks (for example, via session hijacking) won't be able to penetrate their servers.

A second model of anonymizing is called "onion routing." Developed by researchers at the U.S. Naval Research Laboratory, onion routing is based on the idea of "layering" encrypted messages. Connections are made through machines called onion routers, which allows both the sender and the receiver to remain anonymous. On this account, the network "grid" consists of permanent socket connections. Sequences of routers are defined by the setup, and each router is only known to its immediate neighbors. At each router layer, data is encrypted differently, thus making the data more obscure as the layers deepen. An important difference from basic anonymization is that the business part of the sanitization occurs at the application layer.

Onion routing relies on proxy servers that define sequences of routing nodes. In a sense, one may think of the entire route itself as encapsulated. The onion routing may use any form of encryption and any network protocol so long as the routers agree. The general strategy is that the initiating router adds layers to the onion, while each transmitting router peels off a layer, decrypts forwarding information, and sends on what remains. The most important difference from the first model of 'anonymizer' is that onion routing protects the identity of both the sender and receiver—from snooping and from each other. If there's a major vulnerability, it would lie at the endpoints of the communication as network sniffers could conceivably detect initial connections.

A third anonymizing metaphor is called "Crowds." Developed by Mike Reiter and Avi Rubin at AT&T Research Labs, Crowds works by obscuring the actual source of Internet traffic by burying it in the traffic of a "crowd" of users. Given sufficient geographical diversity, individual HTTP requests just seem to blend in with one another.

Crowds uses a proxy server, JONDO, that runs on the user's local machine. JONDO communicates with a server called a "blender" to request admission to the crowd. If accepted, the blender serves up a token proxy from the entire crowd at random, and uses that for the identification of the communication. The packets are then routed through the crowd to the destination, thereby obliterating all tracks.

There is less encryption than the onion routing system because Crowds uses a single symmetric key that is shared by every node on the path. If Crowds has a vulnerability, it is likely to be in the area of traffic analysis and being foiled by executable content (Active X, Java) malware.

A fourth paradigm is Lucent's Personalized Web Assistant. Formerly known as Janus, LPWA was released in 1997. Renamed ProxyMate.com as a startup venture in spring 1999, the technology was sold to NaviPath, an Internet access solutions provider, in May 2000 and is no longer supported so far as we know.

LPWA was designed to provide client anonymity and unlinkability between different sites. Unlike the other three services we described, LPWA is "stateless"—that is, it does not keep internal records of users and aliases. It is designed in such a way that it can recognize returning traffic without retaining lists of associations of aliases and actual identifiers. This feature, the generation of pseudonym aliases, together with the filtering of HTTP header fields, and the use of indirection, where the TCP connection between user and site passes through proxies, made LPWA an interesting model. Its limitations were in vulnerability to traffic analysis, both in terms of forward tracking and replay attacks.

Our final model is "Safeweb," funded by Voice of America and the CIA venture firm In-Q-Tel. Safeweb touts itself as the "most widely used online privacy service in the world." Safeweb began operation in October 2000, and in August 2001 was licensed to PrivaSec. PrivaSec later suspended its free public access service citing financial constraints in November 2001. Although no longer available, Safeweb does offer some

Further Reading

Anonymizer Web site; www.anonymizer.com

Berghel, H. Caustic cookies. *Commun. ACM 44*, 5 (May 2001); www.acm.org/~hlb/col-edit/digital_village/apr-01/dv_4-01.html.

Berghel, H. Hijacking the Web. *Commun. ACM 45*, 4 (Apr. 2002); www.acm.org/~hlb/col-edit/digital_village/apr-02/dv_4-02.html.

Federrath, H. Privacy, anonymity and unobservability in the Internet; www.inf.tu-dresden.de/~hf2/anon/.

Hintz, A. Fingerprinting Web sites using traffic analysis. In *Proceedings of the 2nd Workshop in Privacy Enhancing Technologies*, Springer Verlag, April 2002; guh.nu/projects/ta/safeweb/.

Kristol, D.M., Gabber, E., Gibbons, P.B., Matias, Y., and Mayer, A. *Design and Implementation of the Lucent Personalized Web Assistant (LPWA).* Information Sciences Research Center, Lucent Technologies, 1998.

The Lucent Personalized Web Assistant. Bell Labs technology demonstration; www.bell-labs.com/project/lpwa/system.html

Martin, D. and Schulman, A. *Deanonymizing Users of the Safeweb Anonymizing Service.* Technical Report 2002-2003. Boston University Computer Science Department, Feb. 2002.

Reiter, M.K. and Rubin, A.D. *Crowds: Anonymity for Web Transactions.* AT&T Labs-Research. Technical Report 97-115, DIMACS August 1997.

Onion routing Web site; www.Onion-router.net.

Syverson, P.F., Reed, M.G., and Goldschlag, D.M. *Anonymous Connections and Onion Routing.* Naval Research Lab, June 2, 1997; citeseer.nj.nec.com/syverson97anonymous.html.

interesting insights regarding the development and implementation of an anonymous service.

Safeweb offers a sense of anonymity through the use of URL encryption, much like anonymizer.com. Its architecture uses both SSL and JavaScript to encrypt Web traffic, except that it uses Secure Socket Layer (SSL) or HTTPS. The URL looks something like:

http://www.interestedsite.com ->
https://www.safeweb.com/0/-
0/4101/0011

Once the URL is rewritten in the user's browser, an SSL connection to Safeweb is established, which then finds the requested Web site and returns it to the user. Another interesting aspect of SafeWeb's design was the use of cookies, which are considered very dangerous to overall security because of the amount of information they collect. Safeweb uses a form of cookie generation called the master cookie. The master cookie combines the generic cookie information with Safeweb's proprietary cookie information. Of course this is a potential vulnerability should the master cookie be accessed by unauthorized third parties. Another vulnerability is traffic analysis. Even though the traffic is encrypted, much is revealed by the size of the files and packet header contents.

Conclusion

Anonymization is the business of obscuring the source and perhaps also the destination of network traffic. In this column, we've shown some of the best-known environments for anonymizing the Internet. Anonymizer.com's simple design makes it very popular. Crowds provides an entirely different approach by blending traffic into streams to make tracing difficult. Onion routing obscures traffic by adding multiple layers of encryption. LPWA and Safeweb are still uniquely different in their own ways. In our laboratory, we developed our own 'anonymizer'—Anonymator—from which the screen shots appearing here were generated. No matter which technology is considered, anonymizers attempt to accomplish the same result of sanitizing packets.

Overall, the systems vary in terms of overall anonymity protection. Safeweb and Anonymizer both offer more in terms of user privacy by rewriting most of the content and suppressing certain codes, including Java and JavaScript, but no proof that they offer the user enough protection in terms of connection anonymity. Onion routing and Crowds both provide much better protection in terms of connection anonymity. LPWA was designed as a middleman between all the other services, by providing pseudonymity and filtering. Security is a tradeoff between the user's ability to use services and the protection developed by those services.

Hal Berghel (www.acm.org/hlb) is a frequent contributor to the literature on cyberspace and professor and chair of computer science at the University of Nevada at Las Vegas.
Kim Womack (kwomack@crlmail.unlv.edu) is completing her master's degree in Internet security in the School of Computer Science at the University of Nevada at Las Vegas