Global Information Assurance Certification Paper

IP Fragmentation and Fragrouter

Brad Sanford
December 10, 2000

The IP protocol allows an IP packet to be broken apart into several smaller packets that can be transmitted and reassembled at the final destination. This process is called fragmentation and is an integral part of the IP protocol. IP fragmentation allows IP network traffic to traverse different types of network media with potentially different maximum packet size limits without restricting the IP protocol itself to an arbitrarily low limit on packet size.

While IP fragmentation serves an important role in ensuring the efficient transmission of IP packets across heterogeneous network environments, addressing fragmentation has proven to be rather problematic from a security perspective. Many packet filters, firewalls, network intrusion detection systems, and IP stacks do not adequately address all the nuances of IP fragmentation and reassembly.

You need look no further than Bugtraq to see just how problematic IP fragmentation has been. Industry giants like Microsoft, Internet Security Systems, Checkpoint, Cisco, Network Flight Recorder, AbirNet, most Unix vendors, and many distributions of Linux have all encountered significant problems with IP fragmentation over the last couple of years. Many of these fragmentation handling weaknesses have historically manifested themselves as exploitable denial of service vulnerabilities. Exploits like teardrop and jolt2, as well as many others, actively take advantage of these weaknesses to create a denial of service condition within the target host. Unfortunately, fragmentation vulnerabilities have not been limited to denial of service weaknesses. For example, it was recently discovered that one VERY popular firewall product would actually allow fragmented IP packets to leak through the firewall under certain circumstances.

More recently, using fragmentation as a mechanism of obfuscating an attack or avoiding detection by network intrusion detection systems has been gaining in popularity. It is worth noting that nmap the network scanner of choice for security professionals and hackers alike, incorporates a fragmentation scan for just this purpose. Using IP fragmentation for these purposes was elevated to a new level of awareness, however, with the publication of Thomas Ptacek and Timothy Newsham's paper, "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection" in 1998. Among other issues, this paper discussed twenty-seven different patterns of IP network traffic that the authors used to test four major network intrusion detection systems, including ISS Real Secure, WheelGroup's (now Cisco) NetRanger, AbirNet SessionWall, and Network Flight Recorder. At the time of their examination in 1998, none of the four network intrusion detection systems effectively addressed the issues posed by IP fragmentation and reassembly. Although this research sent the network intrusion detection vendors scrambling to resolve these issues, it should be noted that several of the issues raised by this paper remain unsolved in popular commercial intrusion detection products even today.

While Ptacek and Newsham's paper served to increase awareness of the issues posed by IP fragmentation, the publication of Fragrouter by Dug Song in 1999 as a network intrusion detection system testing tool represented a major milestone in the practical utilization of IP fragmentation as a means of avoiding detection by network intrusion detection systems.

Fragrouter works by accepting IP packets routed to it by another system, fragmenting those packets according to one of the schemes first described by Ptacek and Newsham and described below, then transmitting the fragmented packets to the target host. The schemes used by Fragrouter to fragment the incoming packets are as follows:

baseline-1:	Send the original data in a single TCP data segment.
frag-1:	Send the original data in a single TCP data segment, which is broken into 8-byte IP fragments and sent in order.
frag-2:	Send the original data in a single TCP data segment, which is broken into 24-byte IP fragments and sent in order.
frag-3:	Send the original data in a single TCP data segment which is broken into 8-byte IP fragments, with one of those fragments sent out of order.
frag-4:	Send the original data in a single TCP data segment which is broken into 8-byte IP fragments, with the next to last fragment sent twice.
frag-5:	Send the original data in a single TCP data segment which is broken into 8-byte IP fragments, sent completely out of order with the next to last fragment sent twice.
frag-6:	Send the original data in a single TCP data segment which is broken into 8-byte IP fragments, sending the marked last fragment before any of the others.
frag-7:	Send the original data in a single TCP data segment which is broken into 16-byte IP fragments, preceding each fragment with an 8-byte null data fragment that overlaps the latter half of it. This amounts to the forward-overlapping 16-byte fragment rewriting the null data back to the real attack.
tcp-1:	Complete a TCP handshake, send fake FIN and RST (with bad checksums) before sending data in ordered 1-byte segments.
tcp-3:	Complete a TCP handshake, send data in ordered 1-byte segments, duplicating the next to last segment of each original TCP packet.

tcp-4: Complete a TCP handshake, send data in ordered 1-byte segments, sending an additional 1-byte segment which overlaps the next to last segment of each original TCP packet with a null data payload.
tcp-5: Complete a TCP handshake, send data in ordered 2-byte segments, preceding each segment with a 1-byte null data segment that overlaps the latter half of it. This amounts to the forward-overlapping 2-byte segment rewriting the null data back to the real attack.
tcp-7: Complete a TCP handshake, send data in ordered 1-byte segments interleaved with 1-byte null segments for the same connection but with drastically different sequence numbers.
tcp-8: Complete a TCP handshake, send data in ordered 1-byte segments, with one segment sent out of order.
tcp-9: Complete a TCP handshake, send data in out of order 1-byte segments.
tcb-2: Complete TCP handshake, send data in ordered 1-byte segments interleaved with SYN packets for the same connection parameters.
tcb-3: Do not complete TCP handshake, but send null data in ordered 1-byte segments as if one had occurred. Then, complete a TCP handshake with the same connection parameters, and send the real data in ordered 1-byte segments.
tcbt-1: Complete TCP handshake, shut connection down with a RST, re-connect with drastically different sequence numbers and send data in ordered 1-byte segments.
ins-2: Complete TCP handshake, send data in ordered 1-byte segments but with bad TCP checksums.
ins-3: Complete TCP handshake, send data in ordered 1-byte segments but with no ACK flag set.
misc-1: Thomas Lopatic's Windows NT 4 SP2 IP fragmentation attack of July 1997.
misc-2: John McDonald's Linux IP chains IP fragmentation attack of July 1998.

One of the most interesting facts about Fragrouter is that it is not an attack tool itself, rather it is an enabling technology that allows other attacks to avoid detection by network intrusion detection systems. For example, Fragrouter could be used to obfuscate a phf attack against a web server, a buffer overflow attack against a DNS server, or any number of other attacks. Fragrouter certainly raises the bar for network based intrusion detection systems. Lets hope the intrusion detection system vendors are up to the task. Like many security related tools, however, Fragrouter can be used to maintain secure networks as well as for more nefarious purposes. Security professionals concerned with the effectiveness of perimeter security and network based intrusion detection systems within their own environment would be well advised to give this utility a closer examination. The ramifications of this utility and others like it are simply too great to be ignored.

[1] Ptacek, Thomas and Newsham, Timothy. "Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection." January, 1998. URL: http://www.robertgraham.com/mirror/Ptacek-Newsham-Evasion-98.html (December 9, 2000).

[2] Song, Dug. "nidsbench, a network intrusion detection system test suite." 1998. URL: http://www.anzen.com/research/nidsbench/ (December 9, 2000).

[3] Song, Dug. "FRAGROUTER(8) manual page." 1999. URL: http://www.anzen.com/research/nidsbench/fragrouter.html (December 9, 2000).

[4] Graham, Robert. "FAQ: Network Intrusion Detection Systems." Version 0.8.3. March 21, 2000. URL: http://www.ticm.com/kb/faq/idsfaq.html (December 9, 2000).

[5] Frantzen, Swa. "Intrusion Detection FAQ." 2000. URL: http://www.sans.org/newlook/resources/IDFAQ/fragments.htm (December 9, 2000).

[6] Ziemba, Paul, Reed, Darren, and Traina, Paul. "Security Considerations for IP Fragment Filtering." RFC1858. October, 1995. URL: http://www.cis.ohio-state.edu/htbin/rfc/rfc1858.html (December 9, 2000).

[7] Lopatic, Thomas, McDonald, John, and Song, Dug. "A Stateful Inspection of FireWall-1." 2000. URL: http://www.phoneboy.com/fw1/docs/bh2000/blackhat-fw1.html (December 9, 2000).

[8] Cole, Eric and Skoudis, Ed. "SANS Computer and Network Hacker Exploits: Step-by-Step, Part 1 and 2." July, 2000.

[9] Internet Security Systems, Customer Care. "Changes to the RealSecure product line." October 11, 2000. URL: http://www.iss.net/customer_care/whats_new/rs_new.php (December 9, 2000)