Graham Dungan
April 19, 2025
UIN: 332001764

CSCE 413: Software Security
Self-Study: PingOfDeath

What is the Vulnerability?

The PingOfDeath vulnerability is a form of Denial of Service attack that leverages an overflow of data in an ICMP (ping) packet. Traditionally, a ping is used to monitor the connection to a service. When a packet is too large, it is fragmented into multiple packets. Fragmented packets will contain a **Fragment Offset** value to indicate the location of the sent data relative to the other fragmented packets. The IP packet will also contain a total length field of 2 bytes, specifying a maximum length of 65,535 bytes. PingOfDeath resides within the methods of how the data residing within these fields is processed.

What was the Root Cause?

The PingOfDeath vulnerability is the manipulation of one of these fragmented ICMP packets. An attacker will create a very large packet such that the length exceeds the 65,535-byte maximum size. Since the data will be fragmented due to its size, routers and machines in between will not notice the size. However, the machine responsible for reassembling the packet will experience a buffer overflow due to the size and, thus, will most likely freeze, crash, or shut down. Since sending an obscenely large packet will shut down any target vulnerable to PingOfDeath, this is classified as a denial of service (DoS) attack.

What is the Extent of its Impact?

PingOfDeath affected any system that poorly implemented IP reassembly. Vulnerable systems include Windows 95, 98, NT, and 3.1, several Linux/Unix systems, Mac OS 7.x/8.x, and some routers.

PingOfDeath saw heavy use as a means of performing DoS attacks. Reported in 1996, services across the world began experiencing attacks and wide-ranging shutdowns of their services. University networks were targeted and saw some disruptions, and a Romanian teenager used PingOfDeath and SYN flooding attacks to disrupt some service providers.

How to Patch it?

The PingOfDeath vulnerability could be patched by changing the way packet sizes are validated. If the total reassembled length of the packet is greater than the maximum size allowed under IP, it should be dropped. Following, vendors have also included checks to discard fragmented packets that would exceed the maximum size and to drop malformed/overlapping fragments. Since it is a simple buffer overflow, the solution for this vulnerability was ultimately more bounds checking and error handling. It has been agreed that most devices created after 1998 are resistant to these attacks.

How Could it Have Been Prevented?

This vulnerability had originated from some of the earlier days of the Internet. Directly, this vulnerability was the result of a lack of bounds checking, error handling, and input sanitization. Indirectly, this vulnerability was the result of a lack of 0-trust. Even if it is data sent between machines, it is imperative that it is thought of as malicious, let alone error-prone. An oversized message, even without malicious intent, still could have caught this error. Increasing scrutiny by bounds checking (verification) and error handling (to prevent crashes) would have prevented this vulnerability in the first place.

References

1. Ping of death DDoS attack - Cloudfare
2. Ping of Death (POD) - Imperva
3. Ping of Death Attack: How to Detect and Prevent It - Ascendant
4. Ping of death - Wikipedia
5. Romanian Cracker Takes Down the Undernet - WIRED