DDoS Attack Mitigation: What Avi Vantage Protects Against

Avi Technical Reference (v20.1)

DDoS Attack Mitigation: What Avi Vantage Protects Against

view online

Avi Vantage is the last line of defense for most applications. In most deployments, Avi Vantage is directly exposed to public, untrusted networks. To protect application traffic, Service Engines (SEs) are able to detect and mitigate a wide range of Layer 4-7 network attacks. The following is a list of common denial of service (DoS) attacks and directed DoS (DDoS) attacks mitigated by Avi Vantage.

Attack Layer	Attack Name	Description	Mitigation
Layer 3	SMURF	ICMP packets with the dest IP set as the broadcast IP and the source IP spoofed to the victim?s IP	Packets are dropped at the dispatcher layer if the source or destination IP is a broadcast IP or class D /E IP address.
	ICMP flood	Excessive ICMP echo requests to the victim	ICMP packets are rate limited.
	Unknown protocol	Packets with unrecognized IP protocol	Packets are dropped at the dispatcher layer.
	Tear drop	Exploit the reassembly of fragmented IP packets	Packets are dropped in the protocol stack in the SE if fragment offsets are deemed bad.
	IP fragmentation	Bad fragmented packets	Packets are dropped in the protocol stack in the SE.
Layer 4	SYN flood	Send TCP SYNs without acknowledging SYN acks; the victim?s TCP table will grow rapidly	If the TCP table is being filled with half connections (uncompleted TCP 3-way handshakes), begin using SYN cookies.
	LAND	Same as SYN flood except the source and dest IP addresses are identical	Packets are dropped at the dispatcher layer,
	Port scan	TCP/UDP packets on various ports to find out listening ports for next level of attacks; most of those ports are non-listening ports	Packets are dropped at the dispatcher layer.
	X-mas tree	TCP packets with all the flags set to various values to overwhelm the victim?s TCP stack	Packets are dropped in the protocol stack of the SE.
	Bad RST flood	Send TCP RST packets with bad sequence	Packets are dropped in the protocol stack in the SE if the packet sequence numbers are outside the TCP window.
	Fake session	Guess a TCP sequence numbers to hijack connections	To reduce the chance of success for a fake session attack, the SE uses random numbers for the initial sequence numbers.
	Bad sequence numbers	TCP packets with bad sequence numbers	Packets with sequence numbers outside the TCP window are dropped in the protocol stack in the SE.
	Malformed /Unexpected flood	Unrelated TCP packets after a TCP FIN has been sent	Unexpected packets after the FIN are dropped in the protocol stack in the SE.

	Zero/small window	Attacker advertises a zero or very small window (<100) after the TCP 3-way handshake	If the first TCP packet from the client (after a SYN) is received with a zero or small window, the SE drops the packet and a RST is sent.
	Rate limiting CPS per IP	Connection flood	The rate limits configured in the application profile are applied. (App Profile - DDoS - Rate Limit HTTP TCP)
	SSL errors	Inject SSL handshake errors	SE closes the connection after an error.
	SSL renegotiation	Request for renegotiation after establishing an SSL connection	Client-triggered renegotiation is disabled.
	Request idle timeout	Establishing a connection without sending an HTTP request	The control timeout configured in the application profile is used. (App Profile - DDoS - Post Accept Timeout)
	Size limit for header and request	Resource consumption via long request time	The header-size limits configured in the application profile are used. (App Profile - DDoS - HTTP Size)
	Slow POST	Resource consumption via long request time	The body-size limits configured in the application profile are used. (App Profile - DDoS - HTTP Size)
Layer 7 (HTTP)	SlowLoris / SlowPost	Opening multiple connections to the victim by sending partial HTTP requests	The header and body timeouts configured in the application profile are used.
	Invalid requests	Invalid header, body, or entity in HTTP request	The URI length, header length, and body length limits configured in the application profile are used.
	Rate limiting RPS per client IP	Request flood	The limit configured in the application profile is used. (App Profile - DDoS - Rate Limit HTTP TCP)
	Rate limiting RPS per URL	Request flood	The limit configured in the application profile is used. (App Profile - DDoS - Rate Limit HTTP TCP)