



Tomahawk Test Tool (T3)

Brian Smith (bsmith@tippingpoint.com)

IPS Testing Requirements

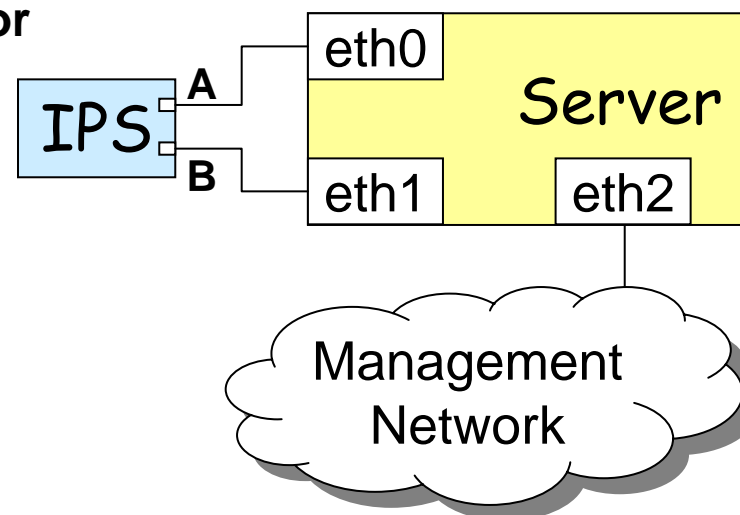
- **Inline Testing**
- **Network Performance**
 - Latency and Throughput
 - Connections and Connections/sec
 - Application Performance
 - Background traffic with realistic protocol mix
- **Security Performance**
 - Coverage? Hard to test
 - Evasion resistance
 - Repeatability
- **The Combination**
 - Blocking under load
 - Throughput/Latency while blocking
 - Throughput/Latency while managing

Open Source Test Tools

- **Tcpreplay**
 - Plays single PCAP multiple times out single interface
 - Performance limited by disk I/O
 - Same 4-tuples replay over and over
 - Can play traffic bidirectionally, but difficult to use
- **Fragroute**
 - Used for evasion testing
 - Works by tweaking routing table to force packets destined for target through loopback interface, where fragroute traps them.
 - Cannot be easily integrated into replay traffic
- **Scanners**
 - Nmap, Nessus, ISS scanner (not free)
 - These make bad test tools, since they mostly do banner scraping

Tomahawk

- **Software running on PC**
 - 3 NICs, 512 MB of RAM, 1 GHz+ processor
- **Tomahawk loads packet trace and replays**
 - Can play hundreds of copies in parallel
 - Maximizes performance, checks for blocking
 - Modifies IP addresses so that each connection is unique
- **Each Tomahawk on 1750 can generate 200 to 750 Mbps**
 - Performance depends on packet size

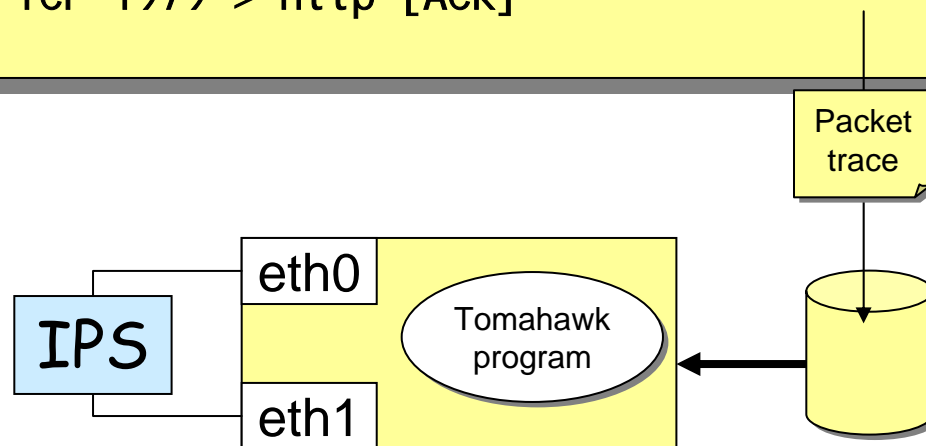


Tomahawk: How it works

```
10.9.103.2 -> 129.27.135.109 TCP 1979 > http [SYN]
10.9.103.2 -> 129.27.135.109 TCP 1979 > http [SYN]
10.9.103.2 -> 10.75.111.181 TCP 1979 > http [SYN]
10.9.103.2 -> 10.75.111.181 TCP 1979 > http [SYN]
10.9.103.2 -> 10.9.103.3 TCP 1979 > http [SYN]
10.9.103.3 -> 10.9.103.2 TCP http > 1979 [SYN, ACK]
10.9.103.2 -> 10.9.103.3 TCP 1979 > http [ACK]
10.9.103.2 -> 10.9.103.3 HTTP GET /scripts/root.exe?/c+di r HTTP/1.0
10.9.103.3 -> 10.9.103.2 HTTP HTTP/1.1 404 Object Not Found
10.9.103.3 -> 10.9.103.2 TCP http > 1979 [FIN, ACK]
10.9.103.2 -> 10.9.103.3 TCP 1979 > http [ACK]
...
```

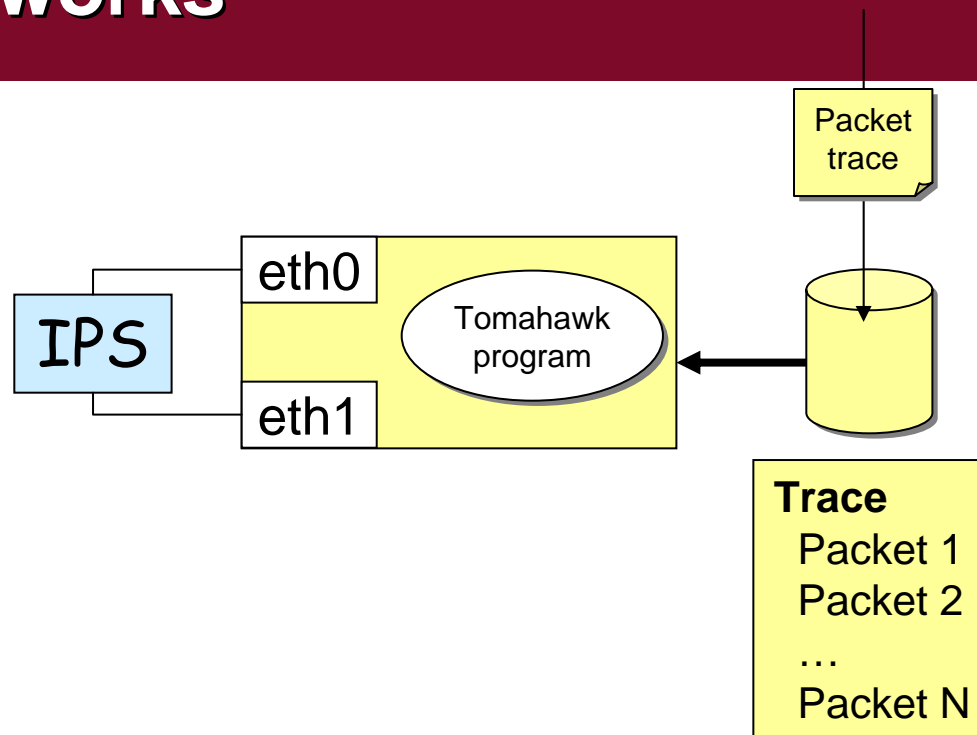
File_IP_Map

<u>IP</u>	<u>interface</u>
10.9.103.2	eth0
129.27.135.109	eth1
10.75.111.181	eth1
10.9.103.3	eth1



Tomahawk: How it works

<u>FileIPs</u>	<u>IP</u>	<u>interface</u>
	10.9.103.2	eth0
	129.27.135.109	eth1
	10.75.111.181	eth1
	10.9.103.3	eth1
	...	



<u>Handler</u>	<u>fileIP</u>	<u>wireIP</u>
	10.9.103.2	10.10.253.1
	129.27.135.109	10.10.253.2
	10.75.111.181	10.10.253.3
	10.9.103.3	10.10.253.4
	...	

What the IPS sees

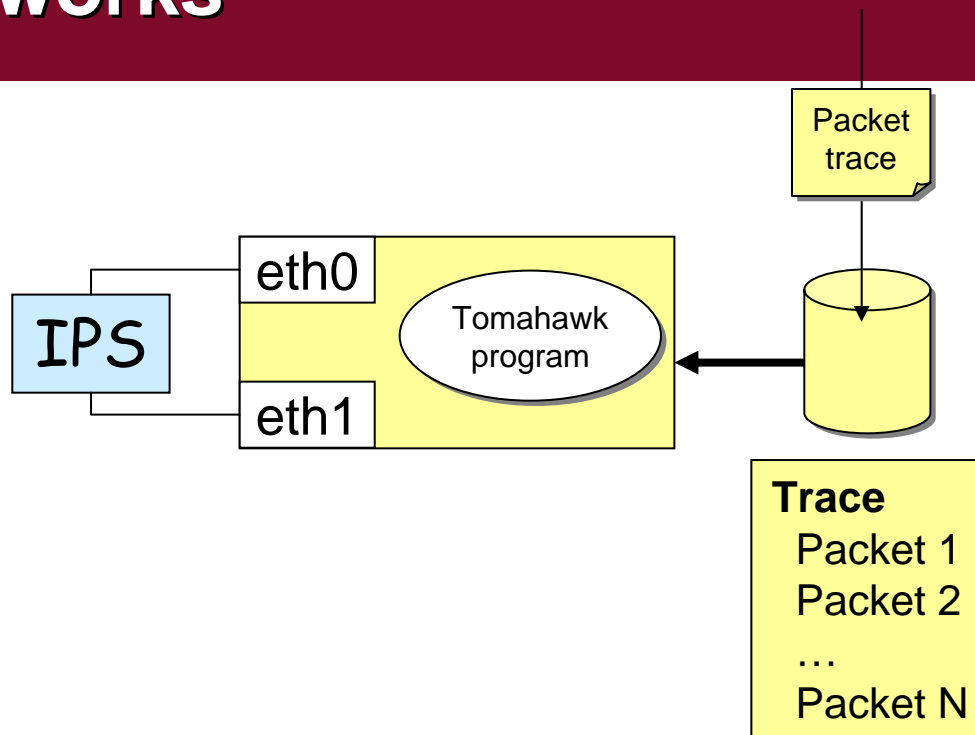
<u>I/F</u>	<u>SourceIP</u>	<u>DestIP</u>	
A	10.10.253.1	-> 10.10.253.2	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.2	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.3	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.3	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [SYN]
B	10.10.253.4	-> 10.10.253.1	TCP http > 1979 [SYN, ACK]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [ACK]
A	10.10.253.1	-> 10.10.253.4	HTTP GET /scripts/root.exe?/c+di r HTTP/1.0
B	10.10.253.4	-> 10.10.253.1	HTTP HTTP/1.1 404 Object Not Found
A	10.10.253.4	-> 10.10.253.1	TCP http > 1979 [FIN, ACK]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [ACK]
...			

Considerations

- **Sending one packet at a time, performance would be limited by latency**
 - Example: 1 second latency, 1 packet per second
- **Pipelining: send several packets at once (e.g., 20)**
 - Example: 1 second latency, 20 packets per second
 - If we send too many, we overflow the receiver NIC queue
 - If we send too few, we do not have good performance
- **Do not want to send responses before requests**
 - Example: if we send SYN, SYN-ACK, but SYN is dropped due to congestion
 - Solution: send only packets going one direction in pipeline
- **Performance drops in “chatty” programs**
 - Solution: send multiple traces in parallel
- **Congestion?**
 - Wait for timeout period (e.g., 100 milliseconds), then resend dropped packets
 - Up to specified number of retransmissions (e.g., 5)

Tomahawk: How it works

FileIPs	
<u>IP</u>	<u>interface</u>
10.9.103.2	eth0
129.27.135.109	eth1
10.75.111.181	eth1
10.9.103.3	eth1
...	



IPmap	
<u>fileIP</u>	<u>wireIP</u>
10.9.103.2	10.10.253.1
129.27.135.109	10.10.253.2
10.75.111.181	10.10.253.3
10.9.103.3	10.10.253.4
...	

IPmap	
<u>fileIP</u>	<u>wireIP</u>
10.9.103.2	10.10.253.8
129.27.135.109	10.10.253.9
10.75.111.181	10.10.253.10
10.9.103.3	10.10.253.11
...	

What the IPS sees

I/F	SourceIP	DestIP	
A	10.10.253.1	-> 10.10.253.2	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.2	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.3	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.3	TCP 1979 > http [SYN]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [SYN]
A	10.10.253.8	-> 10.10.253.9	TCP 1979 > http [SYN]
A	10.10.253.8	-> 10.10.253.9	TCP 1979 > http [SYN]
A	10.10.253.8	-> 10.10.253.10	TCP 1979 > http [SYN]
A	10.10.253.8	-> 10.10.253.10	TCP 1979 > http [SYN]
A	10.10.253.8	-> 10.10.253.11	TCP 1979 > http [SYN]
B	10.10.253.4	-> 10.10.253.1	TCP http > 1979 [SYN, ACK]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [ACK]
A	10.10.253.1	-> 10.10.253.4	HTTP GET /scripts/root.exe?/c+di r HTTP/1.0
B	10.10.253.11	-> 10.10.253.8	TCP http > 1979 [SYN, ACK]
B	10.10.253.4	-> 10.10.253.1	HTTP HTTP/1.1 404 Object Not Found
A	10.10.253.4	-> 10.10.253.1	TCP http > 1979 [FIN, ACK]
A	10.10.253.1	-> 10.10.253.4	TCP 1979 > http [ACK]
...			

Performance and Uses

- **With off the shelf components:**
 - Clean FTP, HTTP, etc: ~300 Mbps
 - Clean UDP: ~200 Mbps
 - Small packets: ~150 Mbps
- **Network Testing**
 - Background traffic
 - Collect trace from target network, replay with Tomahawk
 - Any bottlenecks will show up as performance problem
 - Connections/sec testing
 - Trace with 1000 full TCP connection setup and teardown
 - Each connection: Six 64 byte packets
 - Trace has 6000 packets
 - Replay 250 copies of trace in parallel
 - One PC can do about 31,000 connections/sec



More Uses

■ Security Testing

– Blocking

- Collect trace with attack traffic, replay with Tomahawk
- If trace completes, attack was not blocked

– Repeatability

- Replay, example 20 attack PCAPs with Tomahawk
- Replay each PCAP, example 10 times
- Example total of 200 attacks
- IPS should consistently block or miss all of them
- Most vendors have severe problems with this test!

Tomahawk Jig

- **Application load testing**
 - Use Tomahawk to generate background load
 - NFS copy
 - How long does it take?
 - server₁ to server₂
 - server₁to server₃
 - Recursive HTTP copy
 - How long does it take?
 - Ping can be used to measure latency with and without load
- **Performance while blocking**
 - Use Tomahawk repeatability test to generate attack load
 - Repeat performance tests

