



## **Tomahawk Test Tool Tutorial**

Copyright © 2004 TippingPoint Technologies, Inc., All Rights Reserved.

Unless acquired and licensed from Licensor under another license agreement duly and validly executed by Licensor, the contents of this file are subject to the Reciprocal Public License Version 1.1 (the “RPL”), or subsequent versions as allowed by the RPL, and You may not copy or use this file in either source code or executable form, except in compliance with the terms and conditions of the RPL.

A copy of the RPL can be found in the file titled “LICENSE” distributed with this file or at <http://www.tomahawktesttool.org>.

All software distributed under the RPL is provided strictly on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, AND TIPPINGPOINT TECHNOLOGIES, INC. HEREBY DISCLAIMS SUCH WARRANTIES, INCLUDING WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT OR NON-INFRINGEMENT. See the RPL for specific language governing rights and limitations under the RPL.

# Tomahawk Tutorial

Version 1.0, Sept 30, 2004  
Brian Smith, TippingPoint, Inc.

## Introduction

Tomahawk is a tool for testing the performance and inline blocking capabilities of IPS devices. Tomahawk is run on a machine with three NICs: one for management and two for testing. The two test NICs (eth0 and eth1, by default) are typically connected through a switch, crossover cable, or network based IPS.

**NOTE:** The network connecting the two testing NICs must be a layer-2 network.

Briefly, Tomahawk divides a packet trace into two parts: those generated by the client and those generated by the server. Tomahawk parses the packet trace (called the pcap) one packet at a time. The first time an IP address is seen in a file, the IP address is "assigned" to the client if it is in the IP source address field of the packet, or assigned to the server if it is in the destination field. For example, consider a pcap consisting of a standard three-way TCP handshake that contains three packets:

Packet 1 (SYN):	ip.src = 172.16.5.5 ip.dest = 172.16.5.4
Packet 2 (SYN-ACK):	ip.src = 172.16.5.4 ip.dest = 172.16.5.5
Packet 3 (ACK):	ip.src = 172.16.5.5 ip.dest = 172.16.5.4

When Tomahawk reads the first packet, the address 172.16.5.5 is encountered for the first time in the source field, and the address 172.16.5.4 is encountered for the first time in the destination field. The address 172.16.5.5 is therefore associated with the client, while the address 172.16.5.4 is associated with the server.

When the system replays the attack, server packets are transmitted on eth1, and client packets are transmitted on eth0. To replay the sequence above, Tomahawk begins by sending packet 1 (a client packet) over eth0. When this packet arrives on eth1, it sends packet 2 out eth1 and waits for packet 3 to arrive on eth0. When the packet arrives, Tomahawk sends packet 3 on eth0. When the last packet arrives on eth1, Tomahawk outputs a message that it has completed the pcap.

If a packet is lost, the sender retries after a timeout period (the default is 0.2 seconds). The sender infers that the packet is lost if it does not receive the next packet in sequence within the timeout. For example, if Tomahawk sends packet 2 on eth1 and does not receive it on eth0 within the timeout, it resends packet 2. If progress is not after a specified number of retransmissions, the session is aborted and Tomahawk outputs a message indicating that the session has timed out.

To ensure that the packet is correctly routed through the switches, the Ethernet MAC addresses are rewritten when the packet is sent. In addition, the IP addresses are also rewritten and the packet's checksums updated accordingly. Thus, in the example above, when Tomahawk sends packet 1, the IP source address of the packet that appears on the wire is 10.0.0.1, and the IP

destination address is 10.0.0.2 . Tomahawk writes the modified packet directly to the Ethernet driver using a raw socket.

Within the context of an IPS, if Tomahawk reports that the pcap containing the attack has timed out, the IPS has blocked the pcap. This behavior may be correct, if the pcap contains an attack and the IPS has been configured to block. If Tomahawk reports that the pcap has completed, then IPS missed the attack, regardless of what the log indicates.

## Commands and Examples

The following sections detail some of the commands with examples for Tomahawk.

### Play the pcap File

The following plays the file outlook.pcap once:

```
tomahawk -l 1 -f outlook.pcap
```

The output is displayed below:

```
Beginning test
```

```
Completed 1 loop of trace outlook.pcap
```

```
Finished 1 loops of trace outlook.pcap Completed: 1, Timed out: 0  
Retrans: 0  
Sent: 843  
Recv: 843
```

The first line prints when Tomahawk finishes loading the pcap and begins transmitting.

The second line:

```
Completed 1 loop of trace outlook.pcap
```

prints when Tomahawk completes the replay of the pcap. If the pcap replay did not finish (perhaps because it was blocked by an IPS), the word "Completed" is replaced by the word "Timeout."

The last group of lines:

```
Finished 1 loops of trace outlook.pcap Completed: 1, Timed out: 0  
Retrans: 0  
Sent: 843  
Recv: 843
```

give aggregate statistics for all replays of outlook.pcap. This includes the number of packets sent, number received, and number of retransmissions.

## Play pcap File in Succession

To play this pcap five times in succession, you would use the following:

```
tomahawk -l 5 -f outlook.pcap
```

The "-l" parameter controls the number of loops. This generates the following output:

```
Beginning test
```

```
Completed 1 loop of trace outlook.pcap
```

```
Completed 1 loop of trace outlook.pcap
```

```
Completed 1 loop of trace outlook.pcap
```

```
Completed 1 loop of trace outlook.pcap
```

```
Completed 1 loop of trace outlook.pcap
```

```
Finished 5 loops of trace outlook.pcap Completed: 5, Timed out: 0
```

```
Retrans: 0
```

```
Sent: 4215
```

```
Recv: 4215
```

**NOTE:** The summary statistics indicate that the 5 replays of outlook.pcap finished without being blocked.

If an attack is replayed through an IPS, the IPS should block the attack. Because an IPS often blocks a stream (identified by a host/port quadruple), Tomahawk gives each replay of the attack its own unique host/port quadruple. Assuming that the pcap contains 2 addresses, Tomahawk rewrites the packets so that the first replay of the attack is from 10.0.0.1 to 10.0.0.2, the second replay is from 10.0.0.3 to 10.0.0.4, and so on.

## Start Address Control

You can control the start address with the "-a" flag. For example:

```
tomahawk -l 5 -f outlook.pcap -a 11.0.0.1
```

starts replay attacks at 11.0.0.1. This flag is useful if you are using multiple machines to generate load. A typical usage is embodied in the following snippet of a script:

```
ADDR=$(ifconfig eth0 | grep 'inet addr' | sed 's/\./ /g' | awk '{print $5}')
tomahawk -a 10.$ADDR.0.1 ...
```

The first line extracts the last octet of the IP address assigned to eth0. The second invokes Tomahawk, giving the machine its own block of 16 million IP addresses.

## Replay Packets in Parallel

The example above plays 5 copies of outlook.pcap sequentially. Tomahawk waits for the first replay to complete before sending the second. You can use the "-n" flag to tell Tomahawk to send the replay packets in parallel. For example:

```
tomahawk -n 3 -l 5 -f outlook.pcap
```

This command replays outlook.pcap 5 times, with up to 3 versions running simultaneously. This feature is useful for taking a sample of network traffic captured at comparatively low speeds and "scaling up" the network it represents. For example, suppose you have a trace of traffic from a 100 Mbps network with 500 hosts. By using the "-n 10" flag to Tomahawk, you can simulate a network with 5000 hosts on a gigabit backbone.

You can also use Tomahawk to play multiple attacks simultaneously. For example:

```
tomahawk -n 3 -l 5 -f outlook.pcap -f slammer.pcap -f codered.pcap
```

This command plays up to 3 copies of Outlook, 3 copies of Slammer, and 3 copies of CodeRed simultaneously. In terms of the tool, it plays 9 simultaneous replays in all, 6 of which (Slammer and CodeRed) are attacks. The number of pcaps that can be loaded is limited only by memory.

## Global and Handler Flags

Tomahawk has two types of flags: global flags and handler flags. Global flags affect all pcaps, handler flags affect subsequent pcaps until overridden. For example, consider the following:

```
tomahawk -n 3 -l 5 -f outlook.pcap -n 2 -l 4 -f slammer.pcap -f codered.pcap
```

This command line provides the following information to Tomahawk:

- To play Outlook 5 times, with up to 3 copies running simultaneously
- To play Slammer 4 times, with up to 2 copies running simultaneously
- To play CodeRed 4 times, with up to 2 copies running simultaneously

Up to 7 pcaps and 4 attacks are running simultaneously, and a total of 8 attacks are run.

## Retransmit Lost Packets

As mentioned above, Tomahawk retransmits lost packets. The parameters for retransmission are controlled with the `-r` and `-t` handler flags. For example:

```
tomahawk -l 1 -r 5 -t 1000 -f outlook.pcap
```

This command tells Tomahawk to wait (at least) 1000 milliseconds before declaring a packet lost ("`-t 1000`") and to retransmit the packet 5 times ("`-r 5`") before giving up and printing a timeout message.

## Retain IP Addresses Without Modifications

Occasionally, the IP address in the packet is an important part of the attack. For example, some Stacheldraht messages set the IP source address to "3.3.3.3". In this case, you would not want Tomahawk to modify the source address in the packet. The `-A` flag controls whether addresses are modified:

```
tomahawk -l 1 -A 0 -f stacheldraht.pcap
```

Use `-A 0` to prevent Tomahawk from changing IP addresses for subsequent pcaps and `-A 1` to allow Tomahawk to change IP addresses for subsequent pcaps. For example:

```
tomahawk -l 1 -A 0 -f stacheldraht.pcap -A 1 -f outlook.pcap
```

In this example, Tomahawk leaves the IP addresses in `stacheldraht.pcap` unchanged; whereas, it modifies the IP addresses in `outlook.pcap`.

## Generate Clean Traffic

Because Tomahawk only replays pcaps, it can be used to generate clean traffic and test the zero-loss performance limit of a network device. For example, suppose the file `http.pcap` contains clean HTTP traffic. The following command generates a large amount of clean traffic:

```
tomahawk -n 50 -l 10000 -f http.pcap
```

**NOTE:** In practice, Tomahawk can generate 70 to 500 Mbps on a machine, depending on the platform and pcaps used. For highest performance, TippingPoint recommends two Intel PRO/LAN 1000 NICs on a 2.0+ GHz Pentium family processor with a server configuration. The Dell 1750 platform also performs well for these needs.

## Limit Data Rate

To limit the data rate generated by Tomahawk, use the `-R` flag. For example, to generate 100 Mbps of clean traffic, use the following:

```
tomahawk -n 50 -l 10000 -f http.pcap -R 100
```

The value of "-R" is a floating point number. To generate 100 Kbps of traffic, you would use the following:

```
tomahawk -n 50 -l 10000 -f http.pcap -R 0.1
```

**NOTE:** This command example may take a while to run.

## Limit Simultaneous Streams

You can load a large number of files and limit the overall number of simultaneous streams as a test. For example, you may want to create an average attack rate of about 10 attacks per second using every one of the thousand attacks in your quiver. Suppose we set the packet timeout to 1 second and allow 5 retransmissions. Each attack tacks 5 seconds to the timeout. To achieve the desired attack rate, you must play 50 attacks simultaneously. The "-N" flag controls this variable:

```
tomahawk -N 50 -l 1 -f attack1.pcap -f attack2.pcap ... -f attack1000.pcap
```

The -N flag differs from -n in that -n is a handler flag and -N is a global flag. This means that -N limits the total number of pcap instances that can be played simultaneously; whereas, -n limits the number of instances of each pcap that can be played simultaneously. For example:

```
tomahawk -N 50 -n 5 -l 10 -f attack1.pcap -f attack2.pcap ... -f attack1000.pcap
```

This command sets Tomahawk to play 5 copies each of attack1 ... attack1000 simultaneously (5000 total). The -N 50 flag sets Tomahawk to pay only 50 simultaneously, not 5000.

## Additional Flags

The following is a list of additional flags:

- h Print help and exit
- q Run in "quiet" mode to reduce output. This is good for background traffic generation
- m Specifies the number of packets to send before reading. This is somewhat like the window size in TCP, but is measured in packets rather than bytes. You can play with this parameter to affect performance. Small values lead to poor performance, large values lead to better performance, but potentially higher loss rate. Higher values still lead to high loss rates, hurting performance. A range of about 20-30 seems to work well.
- n Limits the number of instances of each pcap that can be played simultaneously
- i Specifies the interface to send client packets (default is eth0)
- j Specifies the interface to send server packets (default is eth1)

For a full list of flags, see tomahawk.1 in this directory.