



Cleaning Packet Captures for Network IPS Testing

by

Jack Walsh
Intrusion Detection and Prevention Program Manager
ICSA Labs, a division of Cybertrust Corporation

and

David Koconis, Ph.D.
Technical Lead Network Intrusion and Prevention
ICSA Labs, a division of Cybertrust Corporation

July 12, 2006

Cleaning Packet Captures for Network IPS Testing

Introduction

Replaying packet captures (or traces) taken from corporate networks with Tomahawk version 1.1 is an excellent way to generate background traffic for Network IPS testing. The traffic is not synthetic, not contrived, and is unlimited in terms of the potential protocol mixes. However, until after having undergone a “cleaning” process, packet captures are not fit for use in Network IPS testing. This whitepaper describes the rationale for why packet capture traces need to be cleaned and many of the steps performed to clean them.

Non-Full Frames

The organizations providing packet captures to the Network IPS team used the tcpdump utility to capture the network traffic. In several cases, the utility was not configured to store the full Ethernet frame for all network traffic captured. As a result, several of the packet captures received by the Network IPS team initially contained some non-full frames.

To understand if there are non-full frames in a packet capture, it is helpful to know something about the libpcap file format. Like ethereal and a number of other network sniffers, tcpdump writes packet captures in the format defined by libpcap. The libpcap format is one of the most accepted packet interchange formats. In fact, it has been called, “the ‘common denominator’ for network capture files in the open source world”.¹ The libpcap file format looks like that which is depicted in Figure 1 atop the following page.

Figure 1 shows multiple frames in a packet capture created with tcpdump, each preceded by descriptors indicating the size of the frame as it appeared on the wire and the size of what was actually saved in the capture file. If these sizes are different, then the captured frame that follows is a non-full frame. For example, if the size of the frame on the wire was 1280 bytes but the size of the frame in the packet capture was 1000 bytes then this indicates that 280 bytes of the frame was left out of the packet capture file.

The configuration option for tcpdump that controls how much of the frame will be stored in the file is called the snaplength and, by default, it is set to only 96 bytes. In the example from the preceding paragraph, the snaplength used for the capture was 1000. As a result, any frame larger than 1000 bytes is truncated. To ensure that the full Ethernet frame is captured, the snaplength should be set to 1518.

Attempting to use packet captures with non-full frames introduces problems for testing. One problem is that the Tomahawk replay tool halts upon the first occurrence of a truncated, non-full frame. Even if Tomahawk would have been able to replay such a non-full frame, it is unclear what a particular candidate Network IPS might do when it encounters such a frame. One Network IPS may ignore and pass a non-full frame. Another Network IPS may instead choose to drop such an anomalous frame. In either case, truncated frames may cause undue and additional processing, impacting the candidate’s throughput and latency. For these reasons the Network IPS team removes all non-full frames and their corresponding, associated sessions during cleaning.

¹ Development/LibpcapFileFormat, <http://wiki.ethereal.com/Development/LibpcapFileFormat>

© Cybertrust 2006. All rights reserved. Cybertrust and ICISA Labs are registered trademarks of Cybertrust Holdings, Inc. and/or its affiliates. All other trademarks are property of their respective owners.

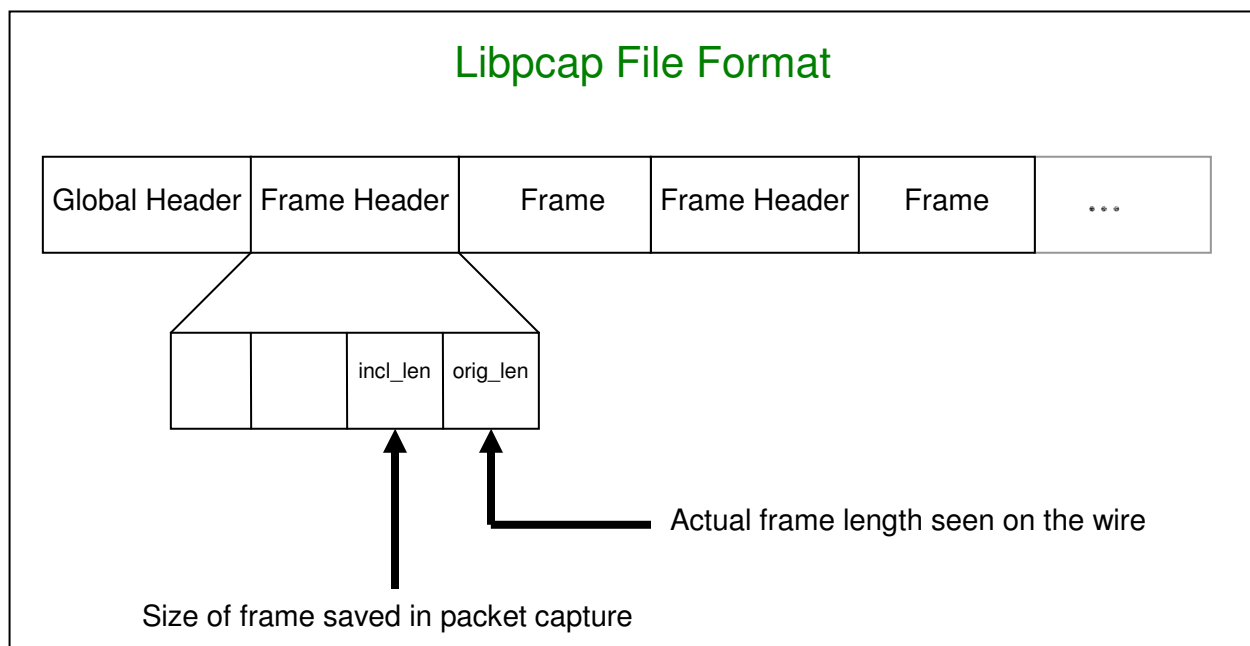


Figure 1

Incomplete Sessions

Packet captures are snapshots of the traffic on a network over a particular period or window of time. It is not uncommon or surprising then for sessions to begin before and/or end after the timeframe during which a trace is captured. Several example sessions in Figure 2 atop the following page have these characteristics.

The figure shows ten sessions either wholly within or partially outside of a window. Everything within the window represents what was captured by a sniffer on a hypothetical network. Everything outside the window was not captured because it occurred at a time when the sniffer was not recording network traffic. From the figure, one sees that session numbers 1, 2, and 3 all started before the packet capture began. Session numbers 2 and 7 completed after the packet capture ended. The remaining six sessions were wholly contained within the trace window.

Because Network IPS devices often keep track of the state of network traffic passing through them, it is unclear what a candidate Network IPS would do with traffic belonging to a session for which there is no beginning (e.g., no three-way TCP handshake). Also, candidates keeping track of state may be unhurried to free up resources for new connections because of the other connections being tracked that were never closed and removed from state tables. And many sessions in a trace without beginnings or endings multiplied by the number of copies of the packet capture being replayed exacerbates any issues the candidate may have associated with resource allocation and clearing.

Further, using packet captures with sessions that begin and end outside of the trace can potentially skew test measurements related to both throughput and latency. Additionally, by tying up resources, replaying packet captures with incomplete sessions could adversely impact

how well candidates mitigate denial-of-service attacks or detect and prevent both attacks and evasions.

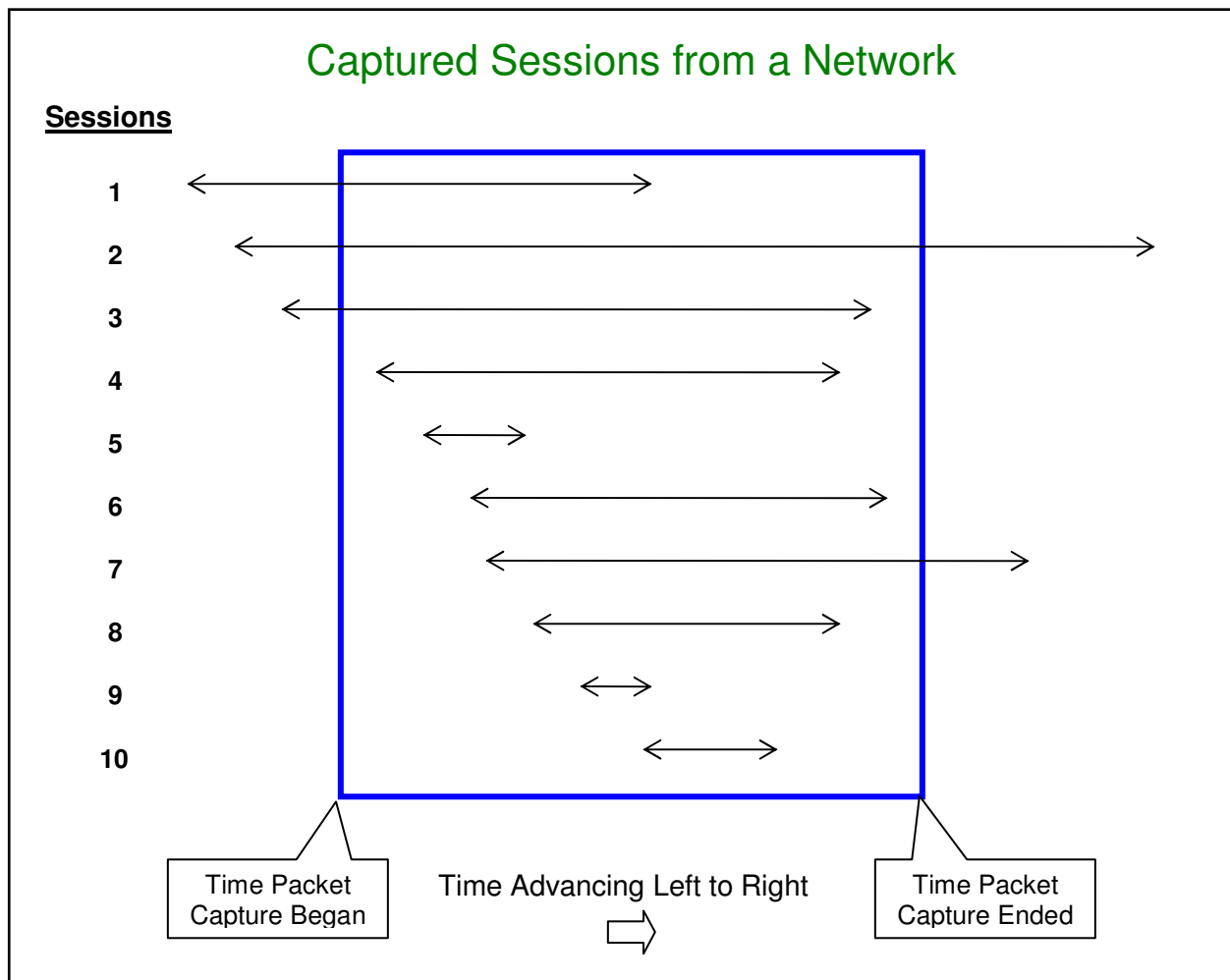


Figure 2

Therefore, the Network IPS team removes all sessions from the original packet capture that begin before and/or end after the time period during which the trace was captured. One should be aware that by removing this traffic, the resultant packet capture may contain a much larger concentration of UDP sessions compared to TCP sessions than the original packet capture. This is primarily due to the fact that TCP sessions are more apt to be removed during this particular cleaning step than UDP sessions, as TCP sessions are typically longer and therefore more likely to begin or end outside the window of time during which a trace was captured.

Malicious Traffic

As it does upon encountering truncated, non-full frames, Tomahawk eventually halts² if a sent frame does not reach its destination. When this occurs it adversely affects testing. It is important then for background traffic to replay continuously throughout testing so as not to halt or unduly burden the Network IPS. Having more than ten Network IPS devices continuously

² This is dependent upon the user-configurable number of retransmissions that is set for Tomahawk.
© Cybertrust 2006. All rights reserved. Cybertrust and ICISA Labs are registered trademarks of Cybertrust Holdings, Inc. and/or its affiliates. All other trademarks are property of their respective owners.

deployed at ICSA Labs – all configured with a policy intended to meet the *Network IPS Corporate Certification Testing Criteria*³ – goes a long way towards helping ensure that any malicious traffic in the packet captures can be identified.

Therefore, prior to testing, the Network IPS team replays packet captures sequentially through all the products continuously deployed in the Network IPS lab – each time stripping out all potentially malicious contents from the trace. The final packet capture to emerge is later used for actual Network IPS testing. It consists of background traffic that is free of malicious, attack traffic. Also, the trace contains nothing other than normal, harmless network traffic that will replay without failure and not unduly slow Network IPS processing. Only the final, cleaned trace is used to ensure the consistency, fairness, and repeatability of ICSA Labs Network IPS testing.

Retransmitted, Duplicate, and Missing Packets

In the real world when a TCP packet does not reach its destination, the sending TCP will wait some time and retransmit the packet. Tomahawk does not take this timing information into consideration. It only cares if the sent packet was received. Once received it sends the next packet in the flow. Since there is little chance of loss, packet captures tend to be replayed much faster by Tomahawk compared to the timeframe over which they were captured. So, if a packet is sent and later re-transmitted and the two packets are seen by the Network IPS within too short a period of time, this may adversely affect packet processing on the Network IPS. Therefore there is a cleaning step to remove retransmitted packets and any related duplicate ACKs.

Other packets that appear in packet captures – like lost segments and ACKs of lost segments – may also adversely affect Network IPS packet processing. In this case the sniffer didn't capture everything that was on the wire, for whatever reason, and now the whole flow in question may look suspicious or create unnecessary issues for the Network IPS. So the offending packet as well as the remainder of the corresponding session are removed from the packet capture during this cleaning step.

Only TCP, UDP, and ICMP

Because Tomahawk can not faithfully replay anything other than TCP, UDP, and ICMP, all other traffic is removed from the packet captures during cleaning.

Post-Cleaning but Pre-Testing

Once all cleaning has been performed, a determination is carefully made as to whether or not the resulting packet capture is actually useful for Network IPS testing. A comparison between the resulting and original packet captures is performed. The following things are among those compared: the percentage of TCP and UDP traffic, the percentage of HTTP and HTTPS to the rest of the application protocol traffic, the number of TCP and UDP sessions, and the change in and the resulting average frame size.

For example, if the average frame size decreased to 200 bytes then the resulting packet capture is probably too unrealistic in terms of the minimum average frame size that one would see on

³ To view version 1.0 of the criteria against which all product to date have been tested, refer to:

http://www.icsalabs.com/icsa/docs/html/communities/nips/criteria/NIPS_criteria_v10_060626.pdf

© Cybertrust 2006. All rights reserved. Cybertrust and ICSA Labs are registered trademarks of Cybertrust Holdings, Inc. and/or its affiliates. All other trademarks are property of their respective owners.

the Internet. Likewise, if the original trace contained 80% TCP packets and 20% UDP datagrams, but the resulting packet capture is comprised of 50% TCP packets and 50% UDP datagrams, the final trace may not accurately represent the original traffic enough to be realistic and useful. Similarly, if the percentage of UDP flows has greatly increased while the percentage of TCP flows has greatly decreased the resultant trace may not be a useful packet capture for testing.

In the last case, a systematic approach can be used to strip out some of the UDP flows without dramatically affecting the distribution of UDP flows over the course of the trace. If additional modifications like this are made, then the resultant packet capture would remain balanced in terms of the number of sessions for each transport layer protocol and continue to have a similar distribution of UDP flows when compared to the original trace.

Summary

ICSA Labs uses real packet captures as background traffic during Network IPS certification testing. The packet captures are taken from live corporate networks and replayed during testing with the ICSA Labs-enhanced version of the open source Tomahawk tool. However, before they can be used in testing, the packet captures undergo a necessary and detailed cleaning process. After passing through the cleaning process each resulting packet capture contains nothing other than TCP, UDP, and ICMP network traffic. Further cleaning removes all non-full frames, malicious traffic, missing packets, and their corresponding sessions. Finally, all incomplete TCP sessions and both retransmitted and duplicate TCP packets are removed. The resulting packet capture is then analyzed, and a determination is made at that point whether or not the packet capture is useful for ICSA Labs Network IPS testing.

Biography

Jack Walsh, Intrusion Detection and Prevention Program Manager

Before coming to ICSA Labs, Jack Walsh worked at the National Security Agency (NSA) for 8 years primarily as an evaluator in the Trusted Product Evaluation Program. While at NSA, he helped author the first-ever US Government Firewall Protection Profile. Since then he has worked 8 years at ICSA Labs. He was the Technical Lead in the Firewall Lab for over 4 years before being named Program Manager for the Network IPS Certification Testing Program. Jack earned his B.S. in Electrical Engineering from The Pennsylvania State University and his M.S. in Computer Science from Johns Hopkins University.

David Koconis, Ph.D., Technical Lead Network Intrusion and Prevention

David Koconis joined ICSA Labs in July 2005. Prior to joining ICSA, he worked at the Institute for Security Technology Studies at Dartmouth College where he managed and was the lead developer of the Security Cybersleuth, a web-based service that collects and indexes computer security related information from open sources on the Internet. He is also a co-author of "Securing Linux-A Survival Guide for Linux Security" (ISBN: 0974372773). David earned his B.S. degree in Aerospace and Ocean Engineering from Virginia Tech and his M.S. and Ph.D. in Aeronautics and Astronautics Engineering from Stanford University.

Acknowledgements

The authors would like to acknowledge the ICSA Labs team for reviewing and providing input into the content of this document.

Who To Contact At ICSA Labs

For questions or comments about this paper contact Jack Walsh at jwalsh@icsalabs.com or David Koconis at david.koconis@icsalabs.com. For more information regarding the ICSA Labs Network IPS Certification Testing Program or Network IPS Product Developers (NIPD) Consortium, visit www.icsalabs.com and follow the “Network Intrusion Prevention” hyperlink.

About ICSA Labs

ICSA Labs, an independent division of Cybertrust, Inc., offers vendor-agnostic testing and certification of security products. Hundreds of the world’s top security vendors submit their products for testing and certification at ICSA Labs. The end-users of security technologies rely on ICSA Labs to authoritatively set and apply objective testing and certification criteria for measuring product compliance and reliability. The organization tests products in key technology categories such as anti-virus, anti-spyware, firewall, IPSec VPN, cryptography, intrusion prevention, PC firewall, SSL-VPN, application firewall, anti-SPAM and Wireless LAN. For more information about ICSA Labs, please visit: <http://www.icsalabs.com>.

About Cybertrust

Cybertrust is the global information security specialist, delivering services that secure critical data, protect identities and help customers demonstrate ongoing compliance. Headquartered in Herndon, Virginia, United States, with more than 30 offices around the globe, Cybertrust is one of the world’s largest providers of information security and is recognized as the global market leader in managed security services. For more information, visit www.cybertrust.com.