SHARKFEST '13

Wireshark Developer and User Conference

SEC-4 Trace File Sanitization NG

Jasper Bongertz, Senior Consultant,
CASSIDIAN CyberSecurity

Topics

I. Reasons for Trace File Sanitization
II. Existing Tools
III. Challenges
IV. A new hope
V. The empire strikes back... sorry, got carried away...

Reasons for Trace File Sanitization

- Removing sensitive information from trace files
- Personal privacy
  - User IDs, passwords, IP addresses,...
- Confidential corporate information
  - Network topology
  - Potential choke points (DoS/DDoS)
  - Device & Software version information (CDP, LLDP)
  - Vulnerable protocols

Existing Tools

- Hex Editors
  - Manual editing of files
- Editcap
  - Removing payloads at a specific offset
- Bittwiste, TCPRewrite
  - Command line, designed as packet replay editor tools
- Pktanon
  - Fully automated sanitization, using XML parameter files

Current status of bittwiste

- Current version is 2.0, dated April 2013
- Often requires multiple calls to replace different parts
- Cannot process PCAPng files
- Doesn‘t like VLAN tags and cannot remove them
- Cannot handle IPv6, tunneling, IPinIP

Current status of tcprewrite

- Current stable version is 3.4.4
- Often requires multiple calls to replace different parts
- Can replace IPv6, but not when tunneled
- Cannot process PCAPng format

Current status of pktanon

- Last version is 1.4.0-dev from September 2011
- Can handle embedded layers, like IP in IP
  - Doesn‘t like some tunneling headers
- Fully automated sanitization, little to no manual control
- No support for PCAPng
- Only tool to do defensive transformation

Sanitization Tools for Network Analysts

- None of the existing tools are specifically designed for
  - „Show and Tell“ preparation (Trainings, Presentations)
  - Transmission of sanitized traces to Vendor support
  - Third party troubleshooting
  - the PCAPng file format

Challenges

- PCAPng file format
- Protocol layers can be very complex to sanitze
  - Even common ones when using exotic protocol options
  - Protocols beyond layer 4
- DNS and DHCP are important, but not easy to do
- Replacement processing:
  - Maximum control over replacement settings
  - Minimum manual „per packet“ effort required
- Defensive Transformation

Challenges

- Truncating/removing layers
  - Typical problem when truncating after TCP: Wiresharks TCP expert goes nuts
- Keeping replacements consistent
  - Across multiple files
  - Addresses, truncating/removing layers
- Checksum calculation
  - Bad CRC should stay bad in the sanitized frame

Challenges

- Unusual suspects
  - IP in IP option block
  - IP in IP encapsulation
  - IP in TCP options
  - Tunneling, e.g. GRE, AICCU/Ayiya (sixxs.net)
  - IPv6, especially extension headers
- Let‘s take a quick look at some samples...

Sanitization tool requirements

1. Read and write PCAPng format without information loss
2. Process at least the most important protocols:
   - Ethernet, including VLANs
   - IPv4/IPv6
   - TCP/UDP
   - ARP, ICMP
3. Support arbitrary protocol sequences
   - Ethernet – VLAN – VLAN - IPv4 – UDP – AYIYA - IPv6 – UDP – UDP-Payload

Sanitization tool requirements

4. Sanitize PCAPng block header information
5. Keep replacements consistent
6. Do defensive transformation – if you can‘t parse it, don‘t write it
7. Provide sanitization parameter configuration options in an easy way
8. Process multiple files in one batch, using the same options and replacement values
9. Allow removing VLAN tags

Arbitrary protocol sequence

- Each layer must be parsed, one after the other, from bottom up (e.g. Ethernet -> IP -> TCP...)
- Building the sanitized frame has to happen top down
- All parsed layers must be available when sanitizing any of them
  - E.g. when calculating the TCP checksum you need the TCP payload and the IP pseudo header

A new hope

- Promise made at Sharkfest 2012: a tool to sanitize PCAPng files
  - Let‘s take a look at what I did in my spare time...

TraceWrangler: current status

- TraceWrangler is in „Alpha“ status
  - Which means that it may do things wrong
  - ...which means that I don‘t trust it yet completely
  - ...which means: if I don‘t, you shouldn‘t either ☺
- My advice:
  - Use it, verify if the results are correct
  - ALWAYS keep the original trace (which is a rule of thumb anyway)
  - If you read formats other than PCAPng and it crashes, convert to PCAPng (editcap...) and try again

TraceWrangler: limits

- It can read
  - PCAPng, libpcap, .enc and Sniffer CAP 2.0 formats
  - PCAPng is preferred, all others may or may not work
  - files of 2GB or less, due to the way how files are accessed as memory mapped files.
- It can write
  - PCAPng 1.0 format
  - Period.

TraceWrangler: limits

- It sanitizes
  - Ethernet
  - VLAN tags (also QinQ, QinQinQ, QinQinQinQinQ...)
  - ARP,
  - IPv4
  - IPv6 (no Extensions headers, yet)
  - TCP,UDP
  - AYIYA (www.sixxs.net)
  - ICMPv4 (no ICMPv6, yet)

TraceWrangler: Executable

- Right now: Windows x86 binary only
  - Written from scratch in Delphi XE2
  - Other OS and x64 may happen in the future
- Uses no DLLs, just a stand alone executable
- Writes settings to
  `C:\Users\`*`username`*`\AppData\Roaming\TraceWrangler`
- All settings are stored in SQLite databases
  - Some are in-memory databases for faster access
- Update is simple: replace the old binary

Roadmap

- Support more protocols:
  - IPv6 Extension headers, ICMPv6, DHCP, DHCPv6, DNS, GRE, MPLS
  - Others I‘m not yet sure of how to do best (I already hear someone whispering „HTTP, SMB,...“ ☺)
- Improve anonymization task dialog
  - Automatic import of addresses found in traces
  - Easier handling of task files to reuse them
- Getting rid of the 2GB trace file limit
- Improve performance

License & where to get it

- At the moment, Tracewrangler is free to use for anyone
  - and will stay that way for personal use
  - I‘m not sure about commercial use after Alpha/Beta yet
- http://www.packet-foo.com/tracewrangler/
- Download, unzip, execute.

SHARKFEST '13

Wireshark Developer and User Conference

Thanks! Questions?

jasper@packet-foo.com
blog.packet-foo.com
@packetjay