Tackling the Haystack
Tuesday, June 14, 2016
Agenda

1. Haystack?
2. Capture
3. Methodology
4. Tools
5. Demos/Scenarios
What's your Haystack size?
What's your haystack size?

• This?
What's your haystack size?

• This?
What’s your haystack size?

• This?
Haystack size

- Everybody has a different "haystack size"
  - new analysts may find 20 packets too hard to understand
  - experienced analysts can deal with gigabytes of traffic if they have to
- Capture files
  - dealing with a single file vs. dealing with file sets
Example Sets

- **October 2015**: ~300GB
  - Trouble with latency of CAD designing in Citrix sessions

- **November 2015**: ~500GB
  - "see if you can find anything that we can improve/fix"

- **February 2016**: ~600GB (sliced to 256 bytes)
  - Web application trouble with long proxy chain

- **May 2016**: ~4000GB
  - Checking for Indicators of Compromise
Working with the haystack
Reducing the haystack size

**Knowledge is a basic building block:**
- protocol behavior, especially IPv4/6 and TCP/UDP
- application behavior
- user behavior
- typical network & security devices, e.g. firewalls, packet shapers etc.

**Experience is key**
- spot the important stuff faster
- know what you can safely ignore & not waste time on
- Problem: experience is usually gained after you needed it most
Experience vs. Knowledge

- Knowledge allows you to understand the meaning of the TCP packets
- Experience tells you if this conversations is worth mentioning in a analysis report

<table>
<thead>
<tr>
<th>No.</th>
<th>IF</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
<th>Length</th>
<th>Delta Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>192.168.1.1</td>
<td>192.168.20.20</td>
<td>TCP</td>
<td>57094-389 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1</td>
<td>66</td>
<td>0.00000000</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>192.168.20.20</td>
<td>192.168.1.1</td>
<td>TCP</td>
<td>389-57094 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1</td>
<td>66</td>
<td>0.00000000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>192.168.1.1</td>
<td>192.168.20.20</td>
<td>TCP</td>
<td>57094-389 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0</td>
<td>60</td>
<td>0.000022</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>192.168.1.1</td>
<td>192.168.20.20</td>
<td>TCP [TCP Keep-Alive]</td>
<td>57094-389 [ACK] Seq=1 Ack=1 Win=65536 Len=0</td>
<td>60</td>
<td>0.00000000</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>192.168.20.20</td>
<td>192.168.1.1</td>
<td>TCP</td>
<td>389-57094 [ACK] Seq=1 Ack=2 Win=131328 Len=0</td>
<td>60</td>
<td>0.000012</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>192.168.20.20</td>
<td>192.168.1.1</td>
<td>TCP</td>
<td>389-57094 [RST, ACK] Seq=1 Ack=2 Win=0 Len=0</td>
<td>60</td>
<td>0.00000000</td>
</tr>
</tbody>
</table>
The path to experience

• When no/little experience is available, you can still reduce the haystack using knowledge
  • read documentation on protocols, applications, etc.
  • gather information about IPs/Users/Ports involved
  • get detailed problem descriptions, with exact date/time info

• Basically you'll need to "learn on the fly"

• Double check your findings whenever you're not sure
  • if possible, ask experienced analysts for a review
General Best Practises
TCP Sessions vs. Chess Games

• How many chess games can you watch/play simultaneously?
Same problem with TCP Sessions

- Can you keep track of more than one?

<table>
<thead>
<tr>
<th>No.</th>
<th>IF</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
<th>Length</th>
<th>Delta Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>103891</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=13 Ack=0 Win=5888 Len=0 Tsc=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00001</td>
<td></td>
</tr>
<tr>
<td>103892</td>
<td>10.1.0.1</td>
<td>10.2.0.2</td>
<td>TCP</td>
<td>Seq=10119 Ack=3488389 Win=21992 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>103893</td>
<td>10.1.0.1</td>
<td>10.2.0.2</td>
<td>TCP</td>
<td>Seq=10119 Ack=3488389 Win=19768 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00004</td>
<td></td>
</tr>
<tr>
<td>103894</td>
<td>10.1.0.1</td>
<td>10.2.0.2</td>
<td>TCP</td>
<td>[TCP Window Update] Seq=10119 Ack=3488389 Win=35376 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00011</td>
<td></td>
</tr>
<tr>
<td>103895</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=5889 Ack=1460 MSS=1460 SACK_PERM=1 Tsec=2463769 TSecr=0 Ws=64</td>
<td>82</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>103896</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=3800 Ack=2736 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00001</td>
<td></td>
</tr>
<tr>
<td>103897</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>HTTP</td>
<td>GET <a href="http://webserv1/search_files/images.jpg">http://webserv1/search_files/images.jpg</a> JPEG HTTP/1.1</td>
<td>476</td>
<td>0.000175</td>
<td></td>
</tr>
<tr>
<td>103898</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=4737 Ack=1516 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>103899</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=6185 Ack=18496 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00001</td>
<td></td>
</tr>
<tr>
<td>103900</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>HTTP</td>
<td>GET <a href="http://webserv1/search_files/_image.jpg">http://webserv1/search_files/_image.jpg</a> JPEG HTTP/1.1</td>
<td>476</td>
<td>0.000052</td>
<td></td>
</tr>
<tr>
<td>103901</td>
<td>10.20.0.71</td>
<td>10.3.0.1</td>
<td>TCP</td>
<td>Seq=3800 Ack=2736 Len=0 Tsec=2463769 TSecr=2804173</td>
<td>74</td>
<td>0.00001</td>
<td></td>
</tr>
</tbody>
</table>
Discipline is key

- **Never delete original capture files**
  - you must always be able to check findings in filtered/carved results against the "true" capture

- **Document everything!**
  - this is actually very hard to do consistently (because Lazyness)

- **Try to keep filter chains intact**
  - it should be possible to retrace the steps from the original down to the final filtered result
Teamwork

• If possible, add as many analysts to the task as you can afford
  • biggest team I had was 3 experts analysts working on complex projects

• Not all analysts are equal
  • Basic skills are the same (e.g. TCP), but everybody specializes
  • WiFi, VoIP, Virtualization, SMB/CIFS are special topics

• Challenges are an important instrument
  • "Found root cause!" (me) - "What? Can't be!" (Chris)
Focus! Focus! Focus!

• It's easy to get lost in all the packets
  • Interesting/weird/unusual stuff found everywhere

• For really big tasks, a team leader is required
  • assigns tasks to members
  • keeps track of time spent
  • calls/leads status update meetings

• Add some "candyland time" if you can
  • e.g. "everybody has until lunch to do whatever he wants with the packets"
Mission Parameters

• What are you supposed to do?
• What information do you have to limit the scope?
  • IP addresses
  • Protocol ports
  • User names
  • Date/Time
  • Markers
Capture Setup
Capture Setup

- **Obvious things to consider:**
  - time stamp accuracy
  - lost packet ratio

- **Not that obvious, but important for large captures:**
  - enough free storage?
  - fast enough, too?
  - which file format?
File access

- How can the captured data be accessed?
  - during capture?
  - after capture?
- Multiple strategies:
  - via USB1/2/3 port (ouch, meh, yay)
  - via Gigabit (or faster) NIC
  - pulling HDD from capture device
Capture file parameters

• Single file or file set?
  • Single file
    • shows all the details in Wireshark at once
    • may be too big to load though
    • can be cut into file sets using `editcap -c`
  • File Set
    • Size range from 64MB to 512MB are common
    • conversations may span multiple files
Slicing

**Advantages**
- available disk space (well, not really, but it doesn't write as much)
- can help avoiding dropped frames
- privacy concerns can be dealt with (bluntly)

**Disadvantages**
- you're not storing everything on the wire to disk
- if you realize you needed more bytes of a frame you have a problem
- Reassembly/content reconstruction is not possible
Analysis setup
Analysis setup

- **Number of analysts**
  - if more than one, new challenges appear, e.g. how to share captures

- **Number of workstations**
  - more is better, helping with carve jobs

- **Number of harddrives**
  - reading from one, writing to another beats working on a single disk
  - SSDs preferred, but usually smaller than traditional HDDs

- **Number of monitors**
Typical analysis tasks

- **Carve/Extraction Jobs**
  - automated packet extraction from large files / set of files
  - often run for hours/days, depending on files/tools

- **Filtering**
  - manual filtering in Wireshark or other tools
  - only feasible for single files & small numbers of packets

- **Merging**
  - merge carve/extraction results
A few useful tools (1/2)

• Filtering/carving files
  • Wireshark
  • tshark
  • tcpdump/windump
  • TraceWrangler

• Convert/edit files
  • editcap
  • reordercap
  • TraceWrangler
A few useful tools (2/2)

• **Merging files**
  • Wireshark
  • mergecap
  • TraceWrangler

• **Others**
  • pcaptouch
  • ngconvert
  • Network Miner
  • tcpflow
Demo 1 - Carving "Essentials"
Hints for "Essentials" carving

• "Essentials" may vary based on the task at hand
  • usually always involves TCP handshake/teardown, so filter for "tcp.flags.syn==1 or tcp.flags.fin==1 or tcp.flags.reset==1"
  • DNS and ICMP are safe bets, too

• Distribute carve tasks across workstations if necessary/possible
  • requires distributing traces and planning carve jobs first
Demo 2 - 5 Tuple VLAN Carve
Hints for VLAN carving

• Running tshark once per VLAN may take a long time
  • each time tshark has to read all the original files

• Methods to improve performance:
  • disable irrelevant dissectors (double check!)
  • Divide & Conquer
  • e.g. carve VLANs 10,11,12 in one run, 13, 14, 15 in another, then run again on partial files for 10, then 11, then 12, etc.
  • use tcpdump/windump with BPF
Demo 3 - Extracting Frames
Hints for extracting frames

- Adding filters for tons of frames in TraceWrangler is going to be slow
  - that's because the code isn't optimized at all
  - it's on the ToDo list 😊
- The output settings define to what file frames will be written
Demo 4: Conversation Statistics
Demo 5 - Megalodon