

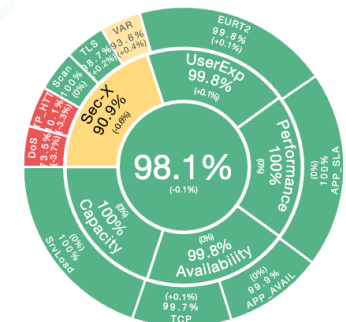


Sharkmon

Monitor your PCAP Data

- anything and everywhere

Andreas Diedrich
Packetfest'25



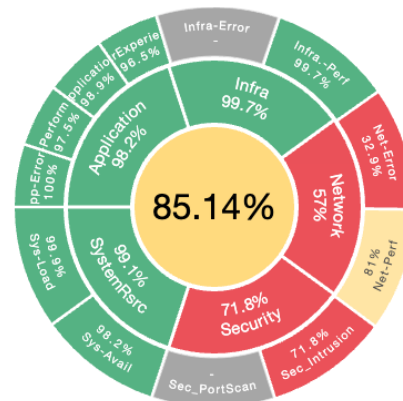
InterView

network solutions
since 1996

DPA
network packet analysis

**IT Service
management**

Since 2015
Working on Integration



Why we are here



Wireshark / Tshark

- Wireshark / Tshark is not just a tool
 - **it is the diamond of packet data**
 - **The global reference for packet analysis**
- 3.000 protocols
- 317.000 display filter = 317.000 metrics !
- Each of these fields including data and/or information
- **Usecases**
 - Network, infrastructure , application, performance, incidents, security, industry, finance, health, Telco...
 - Constantly new protocols
 - (last 4 years - UIC, HTTP/3, MASQUE, STUN, TURN, ICE, SCTP, WebTransport, RIFT, Flowspec v2 amo.)
 - And daily new attack types (ca. 30.000 per year)

Not using tshark for packet analysis monitoring does create immediately gaps



Lets start – tshark for Monitoring

anyone can do it !

Tshark Data delivery

- Each of **317.000 filter fields** can be selected
- **RingBuffer capture** – eg. every minute one file – can cover days, weeks, months, years
- 100s of different data fields in a single command
- various data aggregation schemes (COUNT, MAX, AVG...)
- analysis/inter-frame calculation
 - Retransmissions
 - Timing (RTO, RTT, DNS Time...)
 - experts
- export to standard formats (json, CSV...)
 - Every minute one
 - 1000 Pcap files -> 1.000 Json Files
 - Export DB

Protocol	# Fields
Arp	55
BGP	910
BootP	530
DHCP	591
DLSW	71
DNS	378
ETH	37
FTP	38
H323	45
Hsrp	44
http	107
http2	162
Icmp	108
Imap	18
IP	140
Ipv6	384
Mpls	24
mysql	292
Nbns	52
Netbios	53
Nfs	691
NTP	346
OSFP	337
QUIC	322
...	...

Standard Requirements for DPA Monitoring

- Potentially all Protocols (Tshark logic filter) – user decides !!
- All fields per protocol – user decides !!
- Database for longtime data store
- Incident detection
- Alerting
- Anomaly
- Reporting
- **The rest is “private” decision**
 - Plattform
 - Indexing
 - Dashboards
 - Value



Tshark Commands

1. **Pcap file Summay** - capinfos -TRB trace.pcap
2. **Conversations** - tshark -qr trace.pcap -z conv,ip
3. **Protocols -** - tshark -qr trace.pcap -z io,phs
4. **--z io,stat, data statistisc**
 1. **incl. free selectable granularity, aggregation (avg(MIN/MAX/SUM))**
 2. **tshark -qr trace.pcap -z io,stat,1,"COUNT(tcp)tcp","AVG(tcp.analysis.rto)tcp.analysis.rto"**
5. **„-T fields -e“ every packet data field**
 1. **tshark -2 --enable-protocol transum -qnlr "YOUR_PCAP_TO_REPLACE.pcap" -t u -Tfields -E header=y -e frame.protocols -e frame.time_epoch -e frame.len -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport -e udp.srcport -e udp.dstport -e http.response -e tls.record.version -e tls.record.content_type -e tls.handshake.type -e transum.art -e dns.time -e dns.flags.rcode -e http.time -e http.request.method -e http.response.code -e tcp.flags -e tcp.analysis.zero_window -e tcp.analysis.window_full -e tcp.flags.reset -e ip.ttl -e tcp.flags.fin -e tds.type -e tls.alert_message.level -e tls.alert_message.desc -e tls.app_data_proto -e icmp.code -e smb2.nt_status -e tcp.analysis.rto -e tcp.analysis.retransmission -e smb2.time -e ntp.flags.mode -e ntp.flags.lj -e tcp.analysis.duplicate_ack -e tds.status.reset_conn -e sip.Status-Code -e http.request.uri -e quic.long_packet_type -e quic.ack.gap -e tcp.flags.syn -e tcp.flags.ack -e tcp.analysis.fast_retransmission -e arp.duplicate-address-detected -e dns.qry.name -e tcp.port -e http.request.full_uri -e tls.record.length -e smtp.req.parameter -e ftp.request.command -e tcp.payload -e tcp.analysis.initial_rtt -e tcp.seq -e tcp.ack -e tcp.time_relative -e http.response.code.desc -e http.request -e icmp.type > "YOUR_SCV_TO_REPLACE.csv"**
 - example 125 Metrics
 - creates a record for each packet – huge amount of data !
 - for us biggest challenge / last way

Name	Tshark filter
> HTTP.Response	http.response (cnt, cntdev)
> HTTP.Time (Ind)	http.time (timing, cnt, avg, max, sum)
> HTTP.LOCK	http.request.method == "LOCK" (cnt, cntdev)
> HTTP.GET	http.request.method == "GET" (cnt)
> HTTP.POST	http.request.method == "POST" (cnt, cntdev)
> HTTP.RC500 (Ind)	http.response.code>499 (pct, cnt, cntdev)
> TDS.SQ_Batch	tds.type == 1 (cnt)
> TDS.TDS7_preLogin	tds.type == 18 (cnt)
> TDS.Response	tds.type == 4 (cnt, cntdev)
> TDS.RPC_Call	tds.type == 3 (cnt)

Name	Tshark filter
> HTTP_HTTPS.RC400 (Ind)	http.response.code==400 (pct, cnt, cntdev)
> TLS.v10	tls.record.version == 0x0300 or tls.record.version == 0x0301 (pct, cnt, cntdev)
> TLS.Alert_Warning (Ind)	tls.alert_message.level == 1 (pct, cnt, cntdev)
> TLS.Alert_Fatal (Ind)	tls.alert_message.level == 2 (pct, cnt, avg, max, sum, cntdev)
> TLS.Alerts (Ind)	tls.alert_message.desc (pct, cnt, code, cntdev)
> DoS.Sync (Ind)	tcp.flags == 0x0002 (pct, cnt, cntdev)
> DoS.SynAck (Ind)	tcp.flags == 0x0012 (pct, cnt, cntdev)
> DoS.Final_ACK (Ind)	tcp.seq == 1 && tcp.ack == 1 && tcp.flags == 0x010 (pct, cnt, cntdev)
> Scan.HighSync	tcp.flags == 0x0002 (pct, cnt, cntdev)
> HTTP_HTTPS.SQL-Inject	http.request.uri contains "select" or http.request.uri contains "union" or htt...
> HTTP_HTTPS.XSS	http.request.uri contains "script" (cnt, cntdev)
> HTTP_HTTPS.Pishing (Ind)	http.request.uri contains "php" or http.request.uri contains "login" (pct, cnt, cntdev)
> VAR ARP_Snoop (Ind)	arp.duplicate-address-detected (pct, cnt, cntdev)



Tshark Data commands

What `-z io,stat,`

- Tshark -qr trace.pcap -z io, stat, 10, "COUNT(tcp)tcp", "COUNT(tcp.analysis.retransmission)tcp.analysis.retransmission", "COUNT(tcp.analysis.retransmission)tcp.analysis.fast_retransmission", "COUNT(tcp.analysis.rto)tcp.analysis.rto", "AVG(tcp.analysis.rto)tcp.analysis.rto", "MAX(tcp.analysis.rto)tcp.analysis.rto"
- Benefits : very clear data, data volume is small / each time span one line
- Challenge : no IP addresses / Ports assignment

Who and when requires `-T fields`

- Tshark -qr trace2.pcap -Y "tcp.analysis.retransmission" -T fields -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport -e tcp.analysis.retransmission -e tcp.analysis.fast_retransmission -e tcp.analysis.rto
- **Each packet = one line in export file**
- **10 Gbps = 1 GB/sec = 1 mio pps !!! = 1 Mio lines in export file**

IO Statistics							
Duration: 606.387873 secs							
Interval: 10 secs							
Col 1: COUNT(tcp)tcp							
2: COUNT(tcp.analysis.retransmission)tcp.analysis.retransmission							
3: COUNT(tcp.analysis.retransmission)tcp.analysis.fast_retransmission							
4: COUNT(tcp.analysis.rto)tcp.analysis.rto							
5: AVG(tcp.analysis.rto)tcp.analysis.rto							
6: MAX(tcp.analysis.rto)tcp.analysis.rto							
Interval	1 COUNT	2 COUNT	3 COUNT	4 COUNT	5 AVG	6 MAX	
0 <> 10	199	0	0	0	0.000000	0.000000	
10 <> 20	182	0	0	0	0.000000	0.000000	
20 <> 30	147	0	0	0	0.000000	0.000000	
30 <> 40	181	0	0	0	0.000000	0.000000	
40 <> 50	486	0	0	0	0.000000	0.000000	
50 <> 60	179	0	0	0	0.000000	0.000000	
60 <> 70	143	0	0	0	0.000000	0.000000	
70 <> 80	187	0	0	0	0.000000	0.000000	
80 <> 90	174	0	0	0	0.000000	0.000000	
90 <> 100	223	0	0	0	0.000000	0.000000	
100 <> 110	1828	0	0	0	0.000000	0.000000	
110 <> 120	154	0	0	0	0.000000	0.000000	
120 <> 130	172	0	0	0	0.000000	0.000000	
130 <> 140	852	0	0	0	0.000000	0.000000	
140 <> 150	177	0	0	0	0.000000	0.000000	
150 <> 160	230	0	0	0	0.000000	0.000000	
160 <> 170	513	0	0	0	0.000000	0.000000	
170 <> 180	575	2	0	2	0.021480	0.022695	
180 <> 190	438	7	0	7	0.021444	0.027190	
190 <> 200	106	0	0	0	0.000000	0.000000	

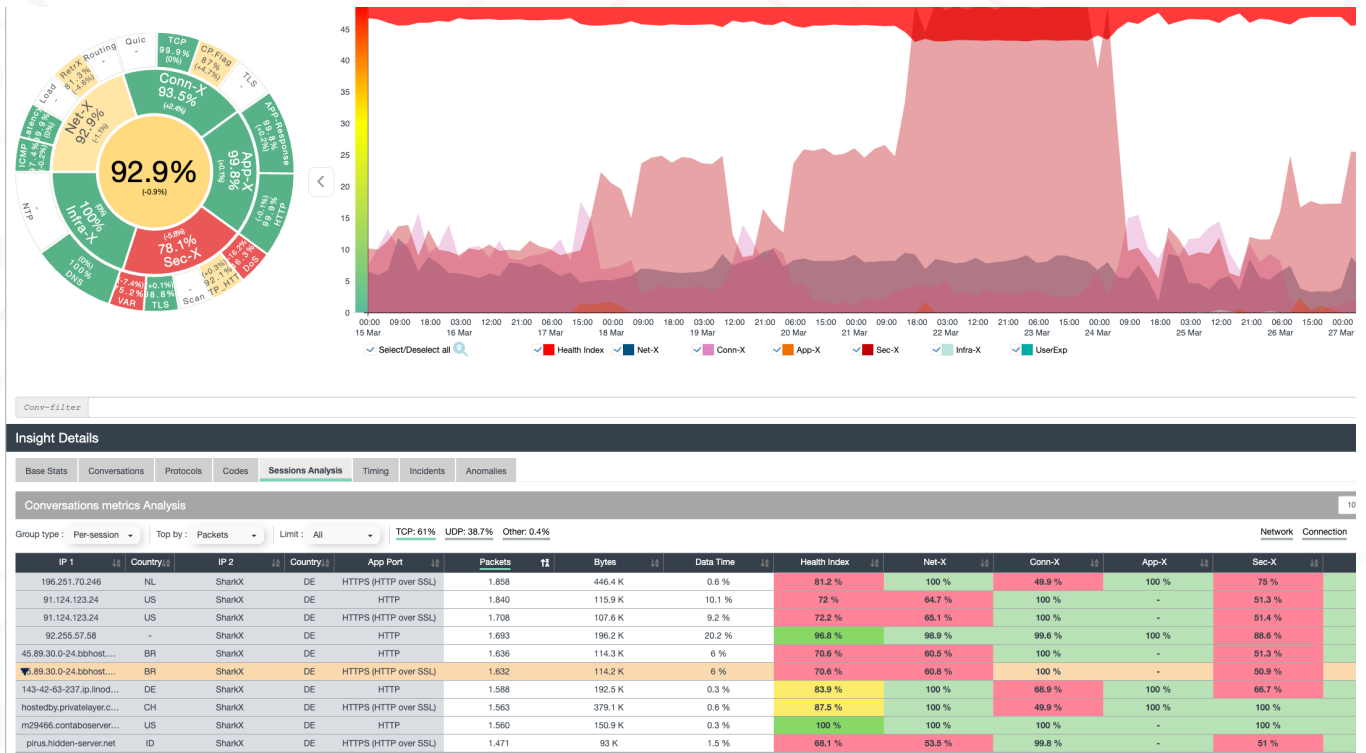
192.168.178.37	17.248.209.72	64256	443	1	0.022825000
13.107.213.45	192.168.178.37	443	64202	1	
192.168.178.37	148.251.120.247	64290	443	1	0.052445000
192.168.178.37	148.251.120.247	64290	443	1	0.092634000
192.168.178.37	148.251.120.247	64290	443	1	0.131608000
148.251.120.247	192.168.178.37	443	64290	1	0.000001000
192.168.178.37	148.251.120.247	64290	443	1	0.079033000
192.168.178.37	148.251.120.247	64306	443	1	0.039949000
192.168.178.37	148.251.120.247	64306	443	1	0.079401000
192.168.178.37	148.251.120.247	64306	443	1	0.067025000
148.251.120.247	192.168.178.37	443	64306	1	0.135705000
192.168.178.37	108.138.36.22	64409	443	1	0.031376000
192.168.178.37	108.138.36.34	64408	443	1	0.044714000
199.232.191.6	192.168.178.37	443	64436	1	



t(shark)mon data and results

- Tshark is the core of analysis
- Sharkmon is the „big data“ organizer – around tshark data

```
tshark -2 --enable-protocol transm -qnlr "YOUR_PCAP_TO_REPLACE.pcap"  
-t u -T fields -E header=y -e frame.protocols -e frame.time_epoch -e  
frame.len -e ip.src -e ip.dst -e tcp.srcport -e tcp.dstport -e udp.srcport -e  
udp.dstport -e http.response -e tls.record.version -e tls.record.content_type  
-e tls.handshake.type -e transm.art -e dns.time -e dns.flags.rcode -e  
http.time -e http.request.method -e http.response.code -e tcp.flags -e  
tcp.analysis.zero_window -e tcp.analysis.window_full -e tcp.flags.reset -e  
ip.ttl -e tcp.flags.fin -e tds.type -e tls.alert_message.level -e  
tls.alert_message.desc -e tls.app_data_proto -e icmp.code -e  
smb2.nt_status -e tcp.analysis.rto -e tcp.analysis.retransmission -e  
smb2.time -e ntp.flags.mode -e ntp.flags.li -e tcp.analysis.duplicate_ack -e  
tds.status.reset_conn -e sip.Status-Code -e http.request.uri -e  
quic.long.packet_type -e quic.ack.gap -e tcp.flags.syn -e tcp.flags.ack -e  
tcp.analysis.fast_retransmission -e arp.duplicate-address-detected -e  
dns.qry.name -e tcp.port -e http.request.full_uri -e tls.record.length -e  
smtp.req.parameter -e ftp.request.command -e tcp.payload -e  
tcp.analysis.initial_rtt -e tcp.seq -e tcp.ack -e tcp.time_relative -e  
http.response.code.desc -e http.request -e icmp.type >  
"YOUR_SCV_TO_REPLACE.csv"
```

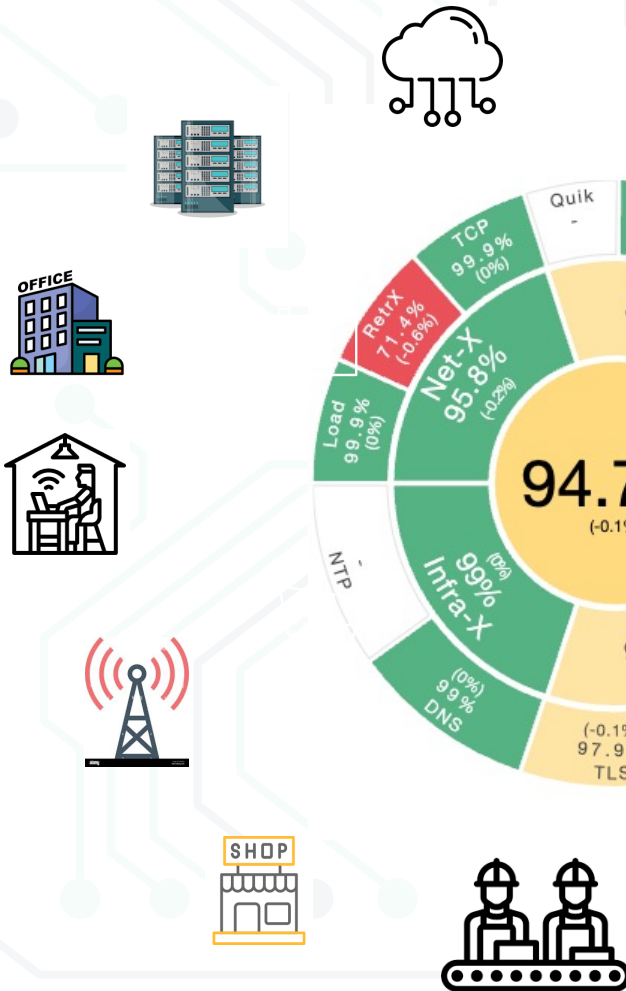




Sharkmon

anything – everywhere

- All known protocols
- Customizable fields
- Customizable indexing



Performance

Network, Server, Application, Infrastructure

Availability

Elements, services

Correctness

Codes, response, behaviour

Security

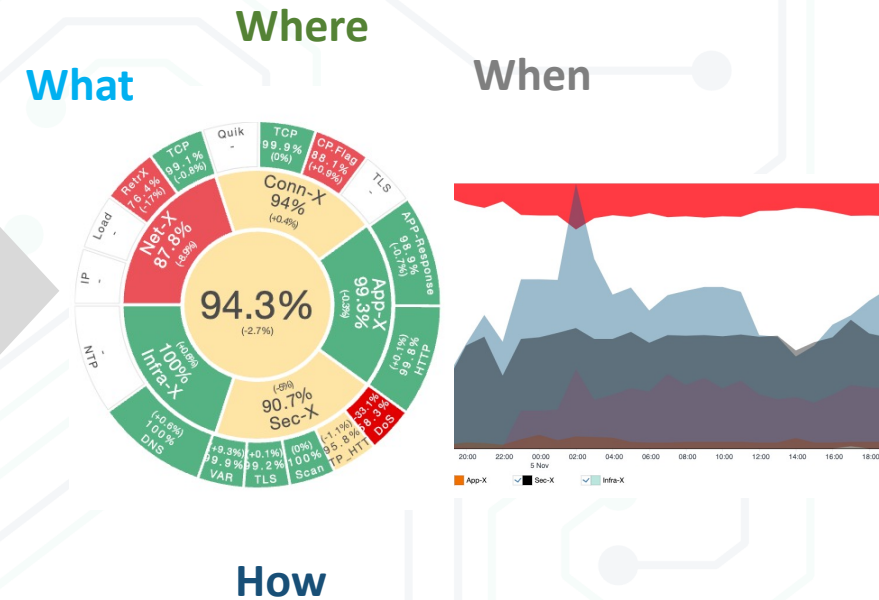
Connections, requests, anomaly



- Calculation
- Identification
- Filter
- Data export



- Data Import
- Indexing
- alerting
- Database





Scenario / Profile

Create new trace scenario

Name: New Scenario

Datasource: Files Drag&Drop

- Files Drag&Drop
- ApplianceApi
- Trace scenario
- SFTP downloading
- Shared folder
- Performance check

Name	ApplianceApi	Default
[M] > SharkMonNew	Files Drag&Drop	Shark
[M] Was_Test	SFTP downloading	Shark
[F] DemoLL	Shared folder	Shark
[F] WAS_DOWNLL	ApplianceApi	Default
[M] SFTP test (inactive)	SFTP downloading	Shark
[F] > SharkXMon	Shared folder	Shark
[F] SecDOS	Files Drag&Drop	Shark
[F] GRE	Files Drag&Drop	Shark
[M] AllegroLL (inactive)	ApplianceApi	Shark

- Multiple parallel scenarios possible
- Multiple different data sources
- Each scenario can use different analysis profile with different categories / classes / thresholds

Trace scenario definition - SharkXMon

Sharing type: Private

Owner: Admin Admin

Name: SharkXMon

Description: SharkX Central

Location Source: Ent

Files control: [hide advanced options](#)

Truncate packets

128 bytes

Search Filter

Define filter before analyze

not espan

Check filter

Apply existing

Enter filter

Split factor

Split by: None

300

Anomalies detection

Type: Past 30 days AVG

HTTP Data

Collect web URLs data

Output files settings

Delete pcap files after analyzing

Export data results

Endpoint

SharkX_2

Save changes

Cancel

Analyzed trace files

Delete selected Reanalyze Download selected Create Sub-Scenario

Trace Id	Created	State	Execution time	Duration, s	Period	Health Index	Sec-X	Conn-X	App-X	Net-X	Intra-X	sharkx_20240923_131122.pcap
91577	23.09.2024	Finished	01:00:26.881	300	23.09.2024 13:26:22 - 23.09.2024 13:31:22	92.9 %	99 %	84.2 %	94.6 %	96.7 %	89.6 %	sharkx_20240923_131122.pcap
91576	23.09.2024	Finished	01:00:15.280	301	23.09.2024 13:21:21 - 23.09.2024 13:26:22	87.9 %	100 %	86.1 %	96 %	96.3 %	67.4 %	sharkx_20240923_132622.pcap
91575	23.09.2024	Finished	01:00:28.451	299	23.09.2024 13:16:22 - 23.09.2024 13:21:21	88.2 %	96.9 %	86.5 %	95.5 %	92.9 %	67.4 %	sharkx_20240923_132122.pcap
91574	23.09.2024	Finished	01:00:22.608	301	23.09.2024 13:11:21 - 23.09.2024 13:16:22	88.3 %	100 %	92.9 %	94.8 %	94.8 %	59.2 %	sharkx_20240923_131122.pcap
91573	23.09.2024	Finished	01:00:29.933	301	23.09.2024 13:06:21 - 23.09.2024 13:11:22	92.3 %	96.5 %	89.3 %	95.3 %	92.1 %	88.5 %	sharkx_20240923_131122.pcap
91572	23.09.2024	Finished	01:00:20.853	301	23.09.2024 13:01:21 - 23.09.2024 13:06:22	92.1 %	98.7 %	81.6 %	94.2 %	92.2 %	93.6 %	sharkx_20240923_130622.pcap
91571	23.09.2024	Finished	01:00:19.301	301	23.09.2024 12:56:21 - 23.09.2024 13:01:22	91.4 %	100 %	92.3 %	94.9 %	93 %	76.7 %	sharkx_20240923_130122.pcap
91570	23.09.2024	Finished	01:00:31.039	301	23.09.2024 12:51:21 - 23.09.2024 12:56:22	88.8 %	100 %	85 %	94.6 %	97.5 %	66.7 %	sharkx_20240923_125622.pcap
91569	23.09.2024	Finished	01:00:36.071	299	23.09.2024 12:46:22 - 23.09.2024 12:51:21	92.1 %	98.5 %	90.3 %	95.2 %	93.1 %	83.6 %	sharkx_20240923_125122.pcap
91568	23.09.2024	Finished	01:00:15.854	301	23.09.2024 12:41:21 - 23.09.2024 12:46:22	90.5 %	97.3 %	86.6 %	94.8 %	93.6 %	80 %	sharkx_20240923_124622.pcap
Total: 91.097												

Update trace profile

Is template

Sharing type: Private

Name: SharkX_2

Description: SharkX Central

Profile metrics

Network (18)

Name	Threshold	Colorize	LED	On Graph	Color	Thresholds	Alerting
> Load.IP	ip (cnt, cntdev)						
> Load.tcpPckt	tcp (cnt, cntdev)						
> Latency.IRTT (Ind)	tcp.analysis.initial_rtt (valdev, avg, max)						
> RetX.RTODelay (Ind)	tcp.analysis.rto (valdev, timing, cnt, avg, max, sum)						
valdev					n/a	V > 0 > V > 0 > V	
timing					n/a	V > 20 > V > 3 > V	
cnt					n/a	V > 7 > V > 5 > V	
avg					n/a	V > 10 > V > 5 > V	
max					n/a		
sum					n/a		
> Load.Packets	ip (cnt, cntdev)						
> RetX.Retransmission (Ind)	tcp.analysis.retransmission (pct, cnt, cntdev)						
> Routing.TTL	ip.ttl (valdev, avg, max, min)						
> Routing.ospf	ospf (cnt)						
> Routing.vrrp	vrrp (cnt)						
> STPSTP	stp (cnt)						

Total: 35

Connection (27)

Application (19)

Security (23)

Name	Tshark filter
> HTTP.HTTPS.RC400 (Ind)	http.response.code==400 (pct, cnt, cntdev)
> TLS.v10	tls.record.version == 0x0300 or tls.record.version == 0x0301 (pct, cnt, cntdev)
> TLS.Alert_Warning (Ind)	tls.alert_message.level == 1 (pct, cnt, cntdev)
> TLS.Alert_Fatal (Ind)	tls.alert_message.level == 2 (pct, cnt, avg, max, sum, cntdev)
> TLS.Alerts (Ind)	tls.alert_message.desc (pct, cnt, code, cntdev)
> DoS.Sync (Ind)	tcp.flags == 0x0002 (pct, cnt, cntdev)
> DoS.SynAck (Ind)	tcp.flags == 0x0012 (pct, cnt, cntdev)
> DoS.Final_ACK (Ind)	tcp.seq == 1 && tcp.ack == 1 && tcp.flags == 0x010 (pct, cnt, cntdev)
> Scan.HighSync	tcp.flags == 0x0002 (pct, cnt, cntdev)
> HTTP_HTTPS.SQL-Inject	http.request.uri contains "select" or http.request.uri contains "union" or htt...

Total: 29

C Distributed Data Import

Monitoring Auto-Import



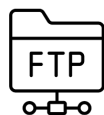
Capture Appliance
API



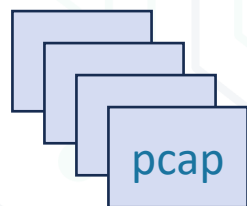
Server OS
Tcpdump/ringbuffer



sFTP auto Import



sFTP Put



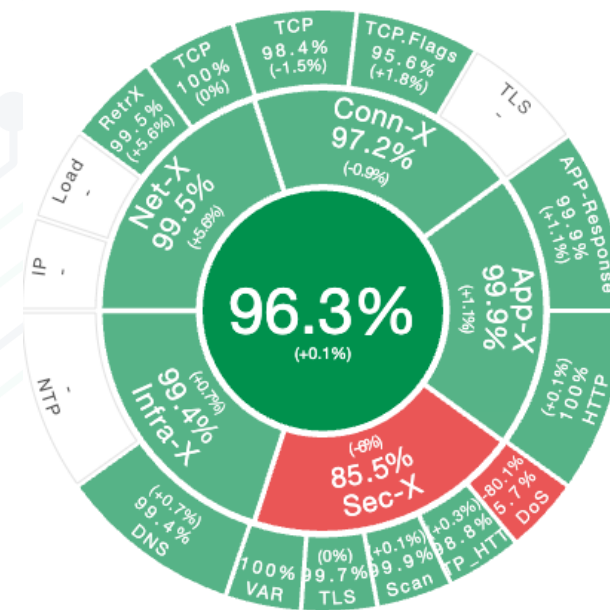
Drag & drop



Manual upload



- Parallel data import
- Multiple datasources
- Multiple locations





t(Shark)mon data

Sharkmon Profile Metrics

- Tshark filter field
- Tresholds
- Naming
- **Category / Class assignment**
- **Indexing**
- **Anomaly**
- MIN/AVG/MAX Values per sec
- COUNT / pSec
- Percent – can use any reference (retrX Per TLS.Sync)
- Codes
 - Tls alert, icmp type, dns response, HTTP reponse – all Codes use
- Timing

Update trace metric

Sharing type: Public

Name: RTODelay

Description: Retransmission Timeout /RTO

Category: Network

Trace class: RetrX

Tshark filter: tcp.analysis.rto

☒ Index configuration

Type: Value

More than

AVG value: 5

IF factor: 1

Proto type: ☒ TCP ☐ UDP ☐ Other

Calculations

Percent: ☐ Select one AGGR: A

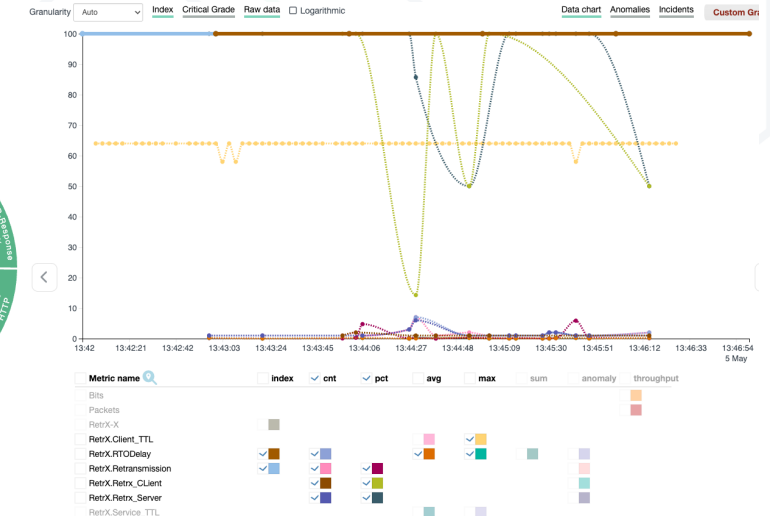
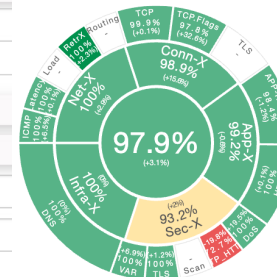
Anomaly: ☐ COUNT ☒ AVG ☐ Percent

Timing: ☒

Code: ☐

Stand-alone: ☒ COUNT ☒ AVG ☒ MAX ☒ SUM ☐ MIN

Round to digit: ☐ 4





Metric what and how

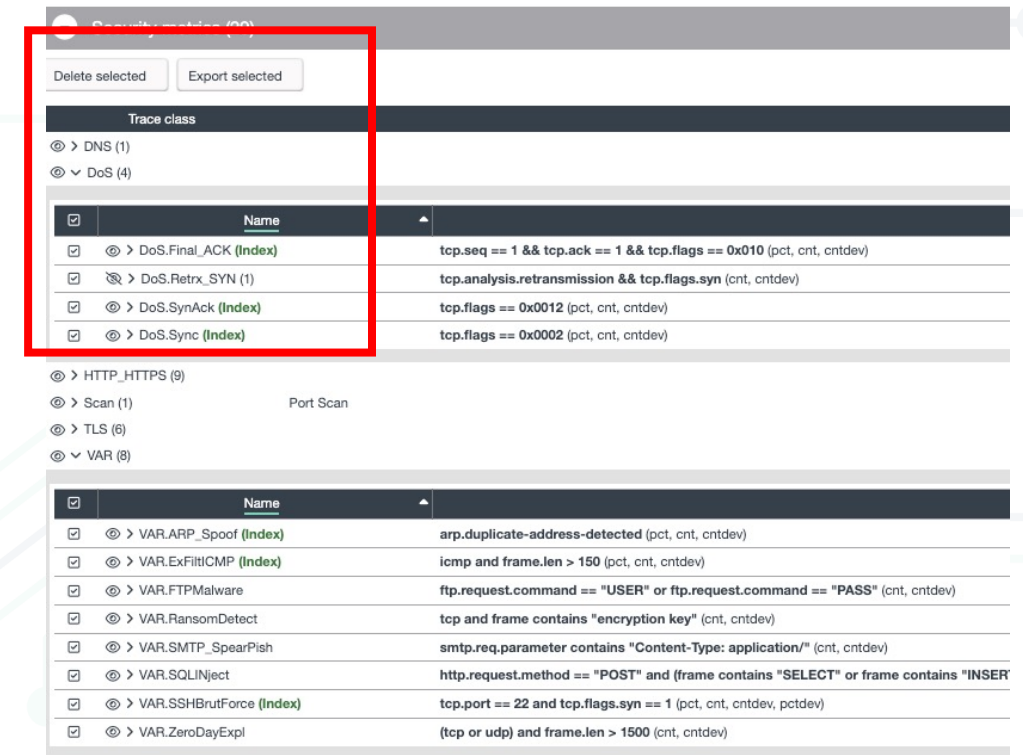
- 317.000 possible fields – nobody can know all !
- So lets use - what **We know** – and **what others know**

Metrics can be shared:

- Export / import

Metric author

- Default set
- User defined own extension
- ChatGPT assistance
- External consultants
- In Future may be reference site



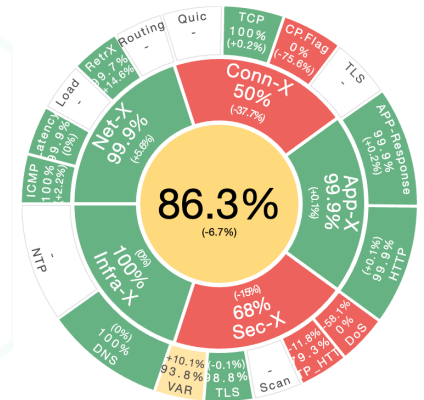


Indexing – why and How AND WHAT !

What is the TOP JOB of Monitoring / analysis ?

- Find the BAD ! – its effects – and ist causes !

- DPA does cover all levels of IT – from most basic transport till final transaction = 7 layers
- **Each layer has often different support team, different skills – and different responsibility !**
- And layers are connected by its effects ! Packetloss & latency may causing bad App responsetime
- Indexing can help indicate
 - what of the much data is critical : the category, the class – and the metric
 - can show clear – who / what was affected !
 - Categories are defined by ist own usecase



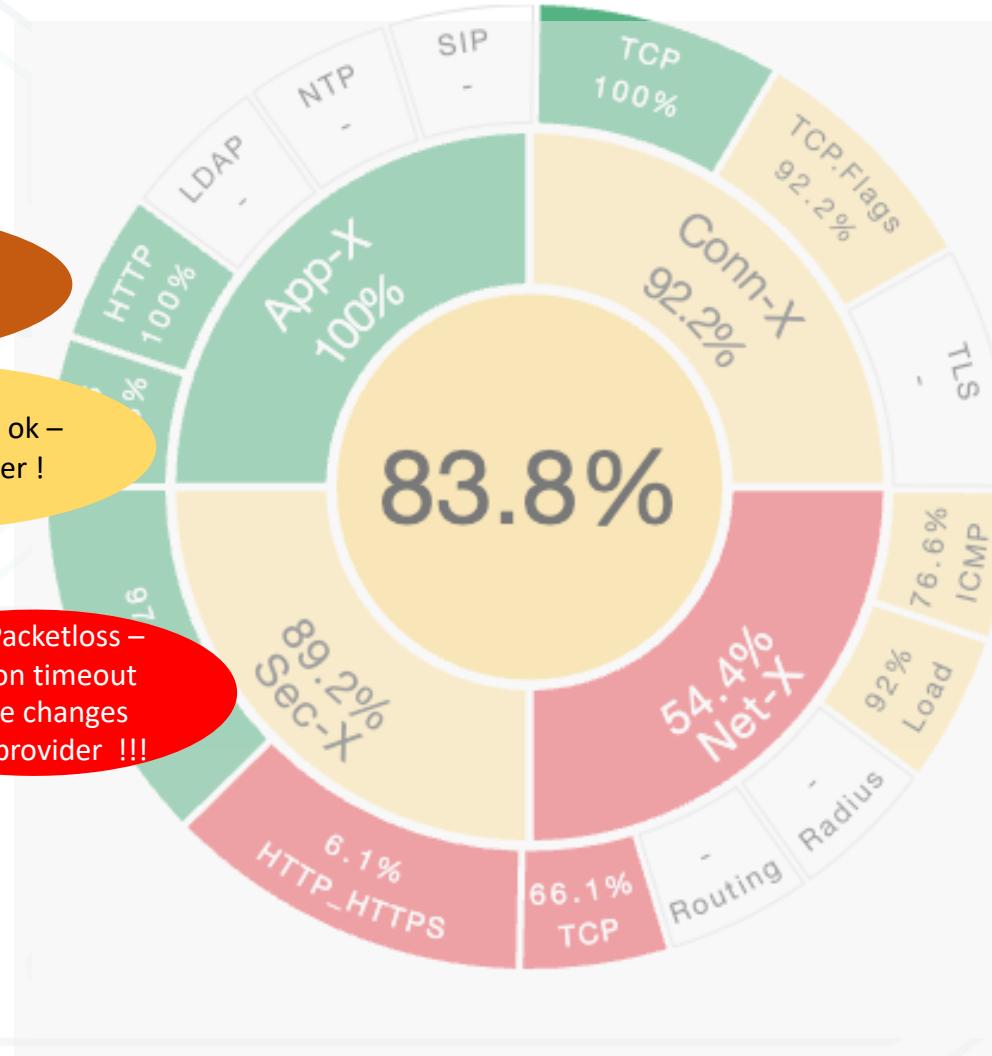
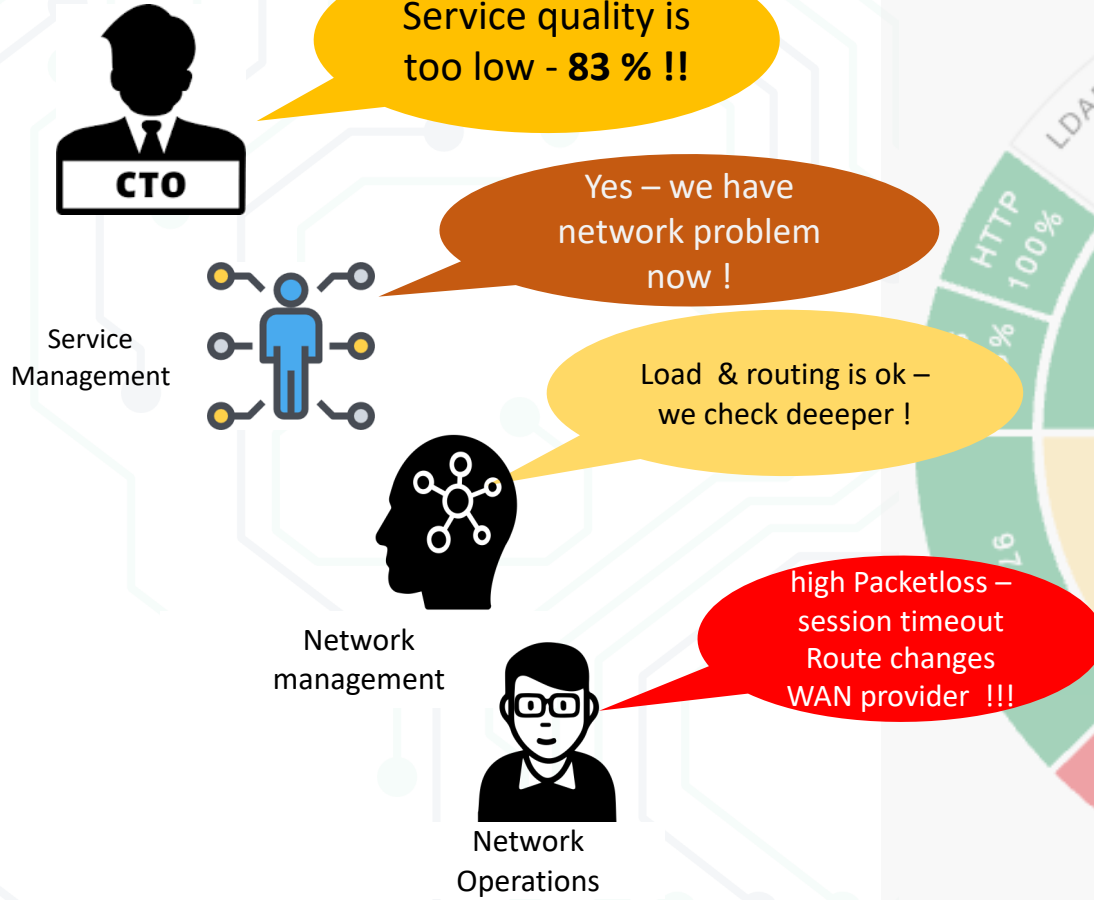
86 % – is this good or bad ?

Indexing value by itself is limited

- 92% today - how was yesterday ? Oh 75% ? We are good !
 - Oh 98 % ? We are bad !
- it needs history – comparison
- And user definable thresholds



Indexing

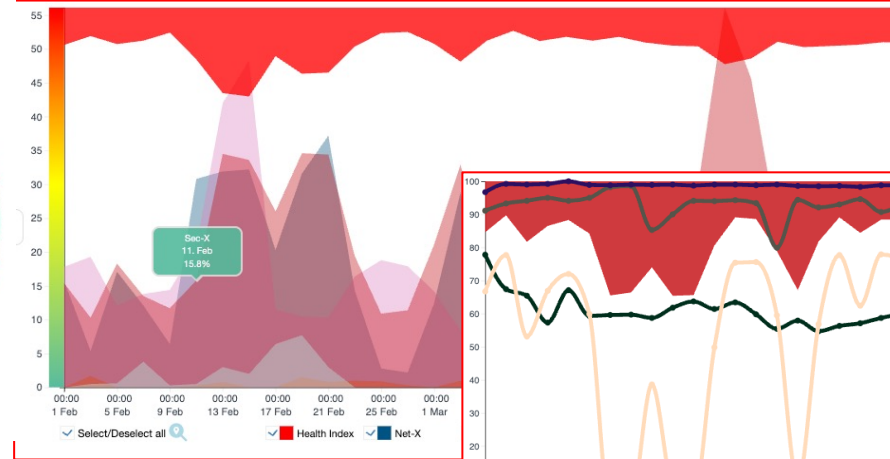
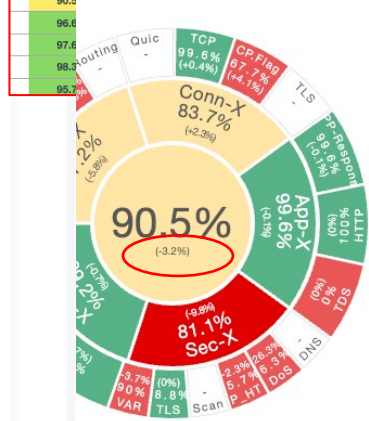




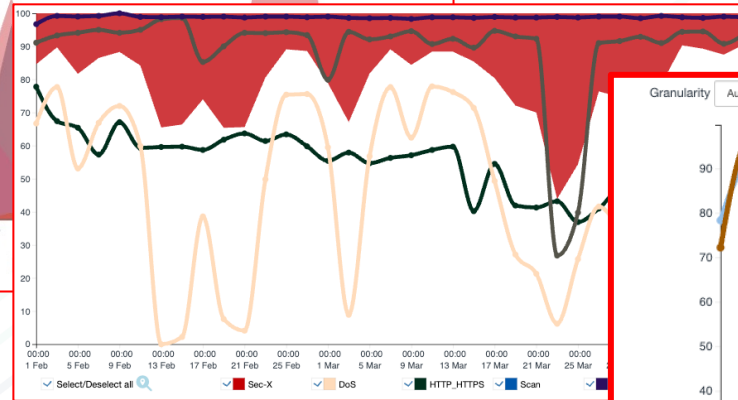
Indexing objects

#	Health Index	Sec-X	Conn-X	App-X	Net-X	Infra-X	File name	File size
1	94.9 %	93.3 %	81.2 %	100 %	100 %	100 %	sharkx_20250506_094628.pcap	75.7 K
2	90.1 %	85.7 %	74.9 %	100 %	100 %	-	sharkx_20250506_094128.pcap	70 K
3	97.5 %	92.3 %	100 %	100 %	97.6 %	-	sharkx_20250506_093628.pcap	67 K
4	91.5 %	92.9 %	74.7 %	100 %	98.3 %	-	sharkx_20250506_093128.pcap	63.1 K
5	90.5 %	85.7 %	74.9 %	100 %	98.3 %	-	sharkx_20250506_093128.pcap	63.1 K

Per PCAP File



Per category / Class



Per Metric



Per Network

Members	IP 1	IP 2	Country	Packets	Bytes	Health Index	Net-X
> 1044	131.0.0.0/8	SharkX	DE	377.769	26.7 M	56.8 %	2.9
> 956	179.0.0.0/8	SharkX	DE	13.182	1.6 M	68.4 %	8.8
> 899	185.0.0.0/8	SharkX	DE	69.641	13.1 M	89.2 %	83.9
> 874	190.0.0.0/8	SharkX	DE	7.861	592.9 K	67.9 %	4.9
> 743	201.0.0.0/8	SharkX	DE	6.634	644.5 K	74.5 %	5.1
> 735	36.0.0.0/8	SharkX	DE	16.912	1.2 M	78.4 %	94.7

Per Conversation

Base Stats

Conversations

Protocols

Codes

Sessions Analysis

HTTP Data

Timing

Incidents

Anomalies

Conversations metrics Analysis

Group type : Per-session

Top by : Packets

Limit : Top 5000

TCP: 83.5%

UDP: 14.7%

Other: 1.8%

Clear

Less than

80

Apply

Cancel

IP 1	Country	IP 2	Country	App Port	Packets	Bytes	Data Time	Health Index	Net-X	Conn-X	App-X	Sec-X
SharkX	DE	ec2-140-1...	CN	SSH/SCP/...	336	22.3 K	2.7 %	58.8 %	73 %	100 %	-	3.4 %
SharkX	DE	150.138.7...	CN	SSH/SCP/...	336	22.4 K	2.6 %	59.1 %	71.5 %	100 %	-	5.8 %
SharkX	DE	150.138.7...	CN	SSH/SCP/...	332	22.1 K	2.6 %	58.5 %	74 %	100 %	-	1.5 %

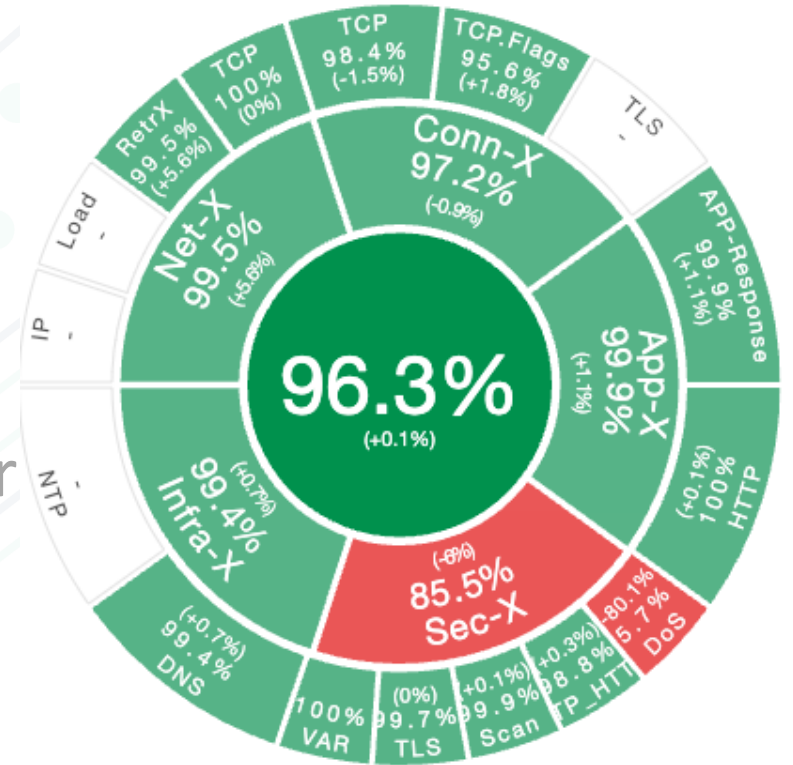


Different Perspectives

the round table view



- Network
 - Server
 - Application
 - Infrastructure
 - Security
- WAN
LAN
Cloud
Provider
SaaS





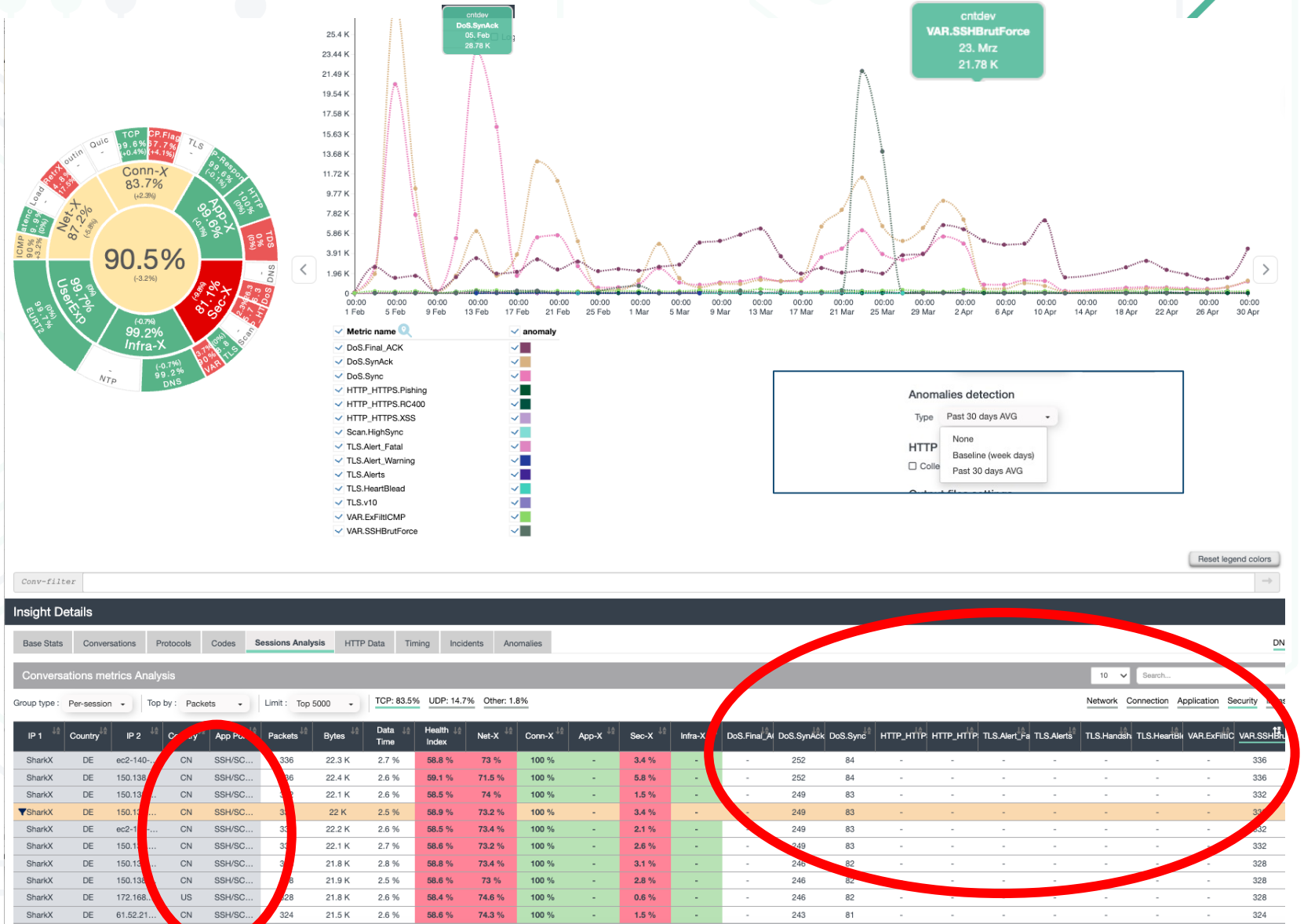
Anomaly – why and how

Problems

- User can not KNOW all metrics
- Fix thresholds often not helpfull
- Cause & Effect often technology-crossing

... and solving

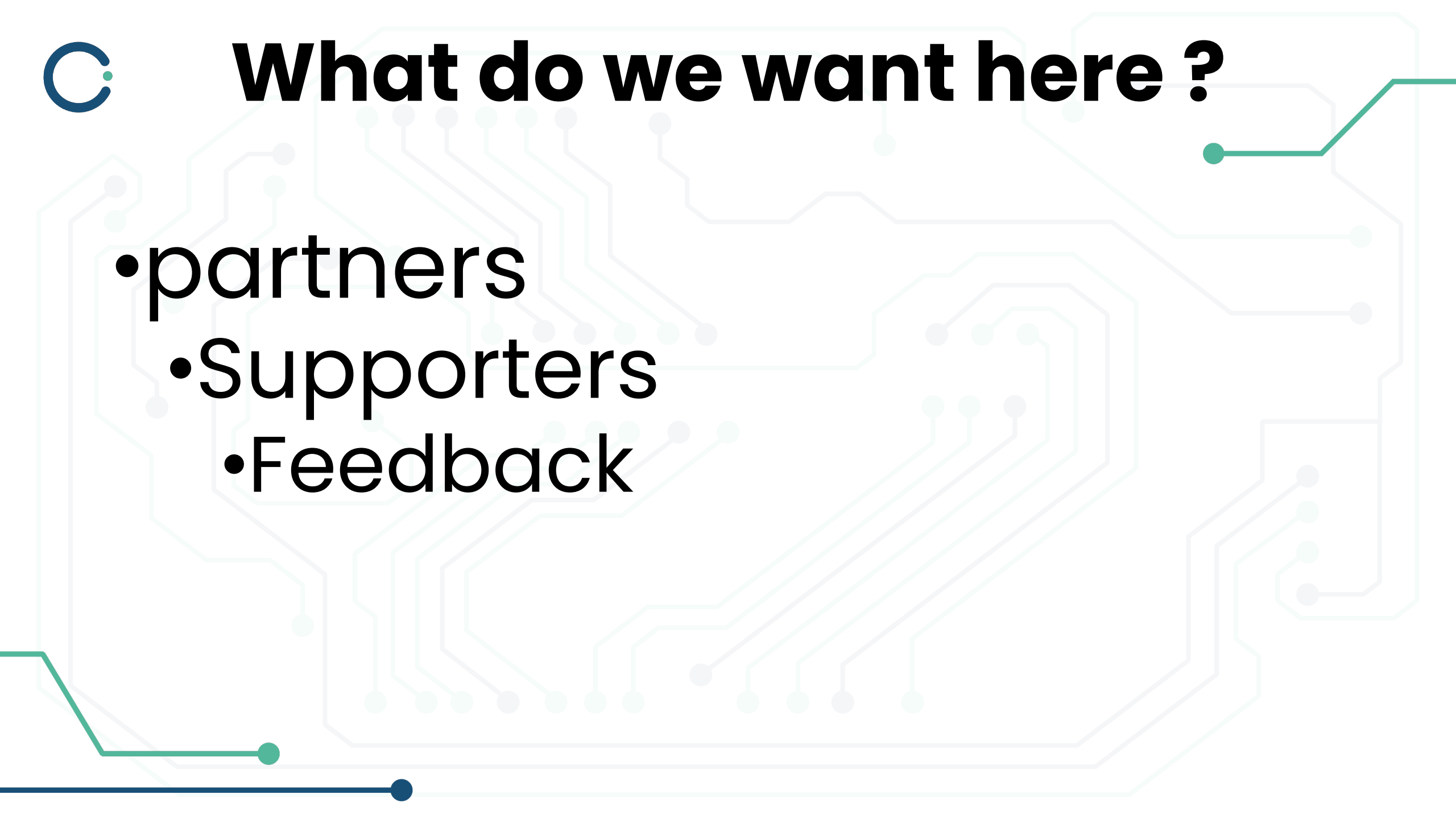
- The anomaly factor – used for many metrics – can help to show heavy changes / **abnormal behaviour**
 - Example: Brutforce : **23.000!! higher than normal**
 - 23.000 remote IPs
- Can show cause & effect – even over different technologies
- We calc constantly AVG over weekday baseline and compare the last seen values with this AVG value



Sharkport

- Easy check features
 - Manual upload files
 - Create / change profiles
 - Invite customers to upload pcaps
-
- Limits
 - no anomaly detection
 - Just manual upload
 - Contact me at ad@inets.de





C What do we want here ?

- partners
 - Supporters
 - Feedback