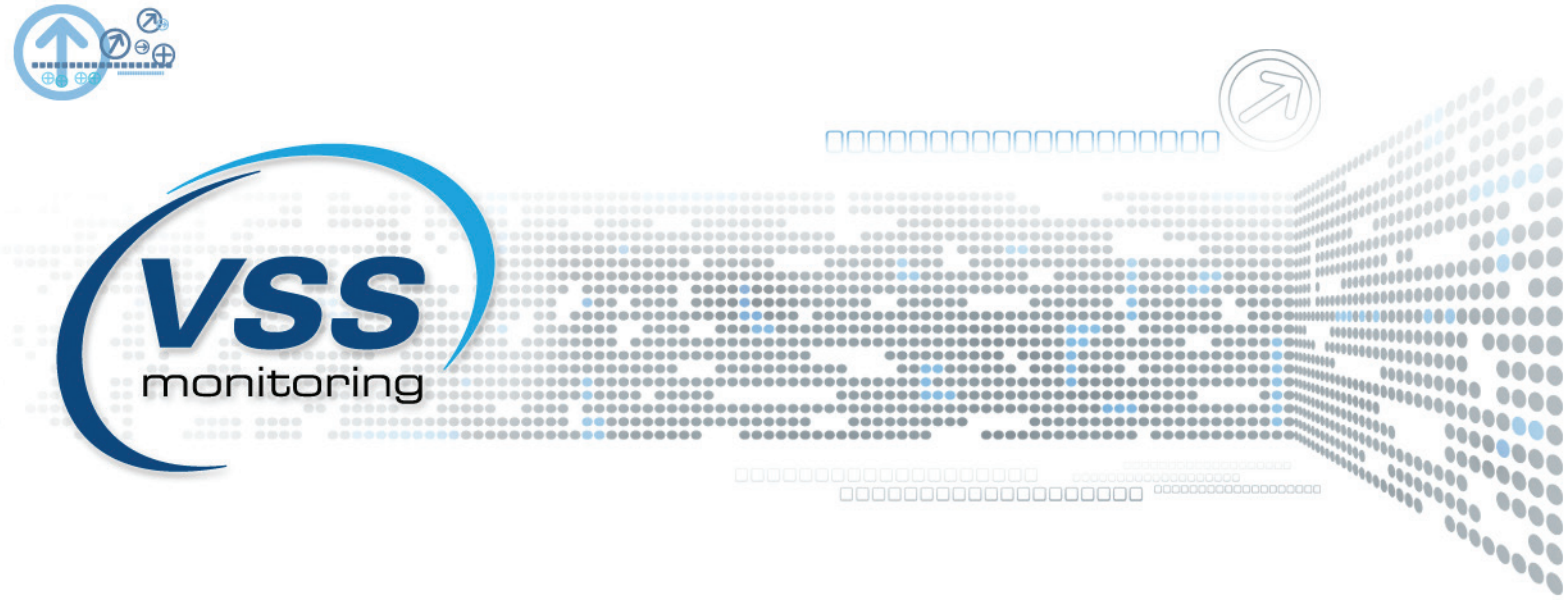


Ramping to a Big Data Visibility Architecture

White Paper

Introduction

Big data is not characterized merely by a large volume of data. Some of the greatest value derived from big data is obtained from the variety of data and the information contained within it. Until big data approaches to analysis emerged, organizations typically had different types of data stored in separate, siloed systems. Big data approaches are often designed to federate fragmented data in order to explore relationships between this data so as to gain valuable insights.

Network data has largely been excluded from big data deployments that work to consolidate and share resources across disparate datasets. This whitepaper makes a case for bringing network data into the big data infrastructure. It goes on to outline potential use cases and practical considerations when designing such a solution.

Why Bring Network Data to Big Data Infrastructure

The network is one of the most important assets a company can have. In some cases, such as with service providers, the network is a fundamental part of the business, and yet network data is mostly excluded from big data systems.

Big data systems gain insights by combining data from different parts of the enterprise which were previously held in separate systems or not stored at all. "Data in motion" across business critical networks represents a major influence on many aspects of the very information big data analytics try to attain. Therefore excluding network data is a missed opportunity to incorporate the significant effect data in motion has on the overall answers analytics are trying to derive.

This section considers some use cases for analyzing network data in big data systems either to gain new insights or to perform tasks which otherwise require purchasing specialized, expensive tools.

Analytics for HFT

High Frequency Trading (HFT) is a type of trading in which market traders profit by using algorithms to trade securities. The strategies deployed for HFT are very sensitive to the latency between (a) the decision to make a trade and (b) actually getting the order into the exchange. Accurately knowing the latency between placement of a trade and arrival at trading venues is absolutely critical.

While there are specialized tools available to calculate this latency in a real time dashboard, historical analysis of latencies at various points in the past are also useful for backward transparency and modeling. For example, it might be worthwhile to investigate the reason for reduced profitability at a certain time in the past. This investigation can be helped by the ability to go back in time and examine exactly what went wrong at that point. Unexpected increase in network latency of a market data feed, a matching system, or a trading venue are all potential problem sources. Other issues leading to packet drops could also lead to increased latency.

Each of these potential problems can be investigated by running analytics on metadata or packet data. Network engineers can analyze timestamps on packets going towards the venue and the TCP ACKs or UDP-transported application ACKs that arrive in response to these packets to receive a clear picture of the latency incurred at a particular point in time. Sequence number analysis of packets can identify TCP/UDP retransmits, which would further identify packets lost at a given point in time. Comparing packet information (e.g. timestamps, sequence numbers) with related market data from the traders systems and the trading venue could also answer questions about latency.

Such analysis, while simple to describe, is extremely compute expensive when performed over large amounts of data. Some specialized tools can perform this analysis; however, due to limited storage and expensive scalability of these tools, it becomes cost prohibitive to store historical data for long periods of time. Bringing this data into big data platforms solves this problem.

Big data analytics platforms are well suited for large scale data processing. Big data platforms also allow network data to be queried in conjunction with data from other parts of an organization such as profit and loss, execution reports, and risk management logs in order to gain better insights into trading activities as a whole.

Analytics for Security

Network security is an increasingly important concern. Attack surfaces are growing more complex: Internal and external actors—many with political and economic motivation—have caused previously unimaginable damage. For this reason, network security tools are one of the chief consumers of network data. These specialized tools correlate data from different parts of the network infrastructure to form a comprehensive picture of its security posture.

Consolidation of information from myriad sources to gain previously unavailable insight is a task that big data platforms do well, and moving security analytics to such a platform (using general purpose computing clusters), can have the added benefit of reducing overall costs and potentially extracting greater insight from the data.

Consider an investigation into whether (and, possibly, what) data was exfiltrated from a company within the past year. Simply by processing flow metrics or metadata from the past year in a MapReduce job, it might be possible to pinpoint anomalously large transfers of data from within the network to outside the network. By creating a baseline for the amount of data transferred by all the network nodes to outside destinations and identifying deviations from this baseline, a subset of network data could be identified for further analysis. Further clues might be yielded by using the metadata to identify the appearances of previously unseen protocols on the network or a departure from the normal baselines for the traffic share of protocols. For example, a sudden surge in the amount of DNS traffic coupled with a suspicious increase in the size of these DNS packets might indicate a DNS tunneling attack taking place at that point in time. Any packet data that is pinpointed by such investigations can then be analyzed (assuming the full packets have been stored) to exactly identify what information, if any, was exfiltrated, by who, when, and to where.

While big data and Hadoop have become almost synonymous, there are also processing paradigms for big data which are based on real time analysis. Thus, security analytics using big data need not only be used for reactive security monitoring but can also be used for real time, proactive identification of threats.

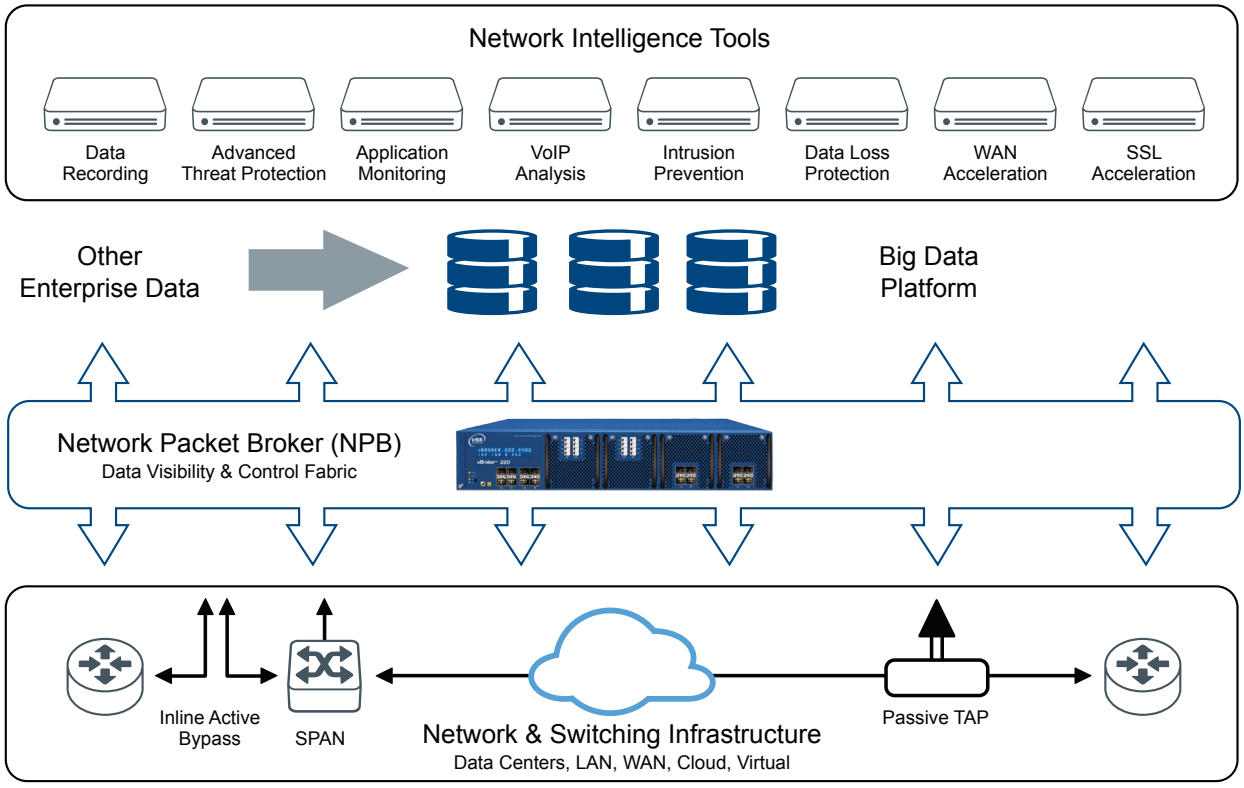
BDV Architecture

The traditional network monitoring architecture has consisted of network taps feeding specialized tools implementing various NPM/APM/Security objectives.

The arrival of Network Packet Brokers (NPB) changed this architecture to include a NPB layer which collected, aggregated, filtered, and optimized data before handing it off to tools. The NPB layer allowed the tools to only see the data of interest to their applications, allowing higher bandwidth network links to be monitored by tools, without needing to scale the tools.

Big Data architectures add a separate layer of consolidated storage into the architecture of network monitoring. Instead of each tool receiving traffic from the network and creating its own copies of data for historical analysis, big data architectures have a centralized storage for network data. This storage is shared for data from other parts of the enterprise.

The following sections introduce some key VSS technologies that facilitate a move towards such a BDV architecture. Some potential use cases are also identified, along with planning considerations.



vSpool

vSpool™ is the VSS Monitoring write-to-file solution that enables network administrators to encapsulate captured network traffic in PCAP file format and spool directly to storage via a network storage protocol, such as File Transfer Protocol (FTP).

The vSpool solution supports the centralized recording of network packets for any single network or group of networks. Captured network traffic directed to a vSpool monitor port can be timestamped, filtered, sliced, de-duplicated, and flow balanced. This ensures that only desired traffic is sent to a storage drive, reducing storage capacity requirements and eliminating concerns over the storage of sensitive data. Preprocessing traffic prior to storage also reduces time-to-retrieval of stored data, as metadata can be applied during encapsulation to establish precise instance and source of packet capture.

vSpool is interoperable with any FTP-capable Network Area Storage (NAS) appliance, offering network administrators significant flexibility in connecting network capture infrastructure to a storage warehouse or content management platform

vIndex

vIndex™ is a solution on VSS NPBs which works in conjunction with vSpool. For every PCAP file that vSpool creates, vIndex creates a tab separated text file containing:

- The Ethernet headers
- The IPv4/IPv6 headers
- The TCP/SCTP/UDP headers
- Timestamp
- Byte pointer to packet within the PCAP file

These vIndex files allow analysis to be run on network data without the need for PCAPs themselves to be processed in the big data systems. All fields that are extracted by vIndex can be directly used in analyses in MapReduce jobs without needing to handle PCAPs.

The byte pointer to the packets within the PCAP file, as discussed in the previous section, makes it easier to retrieve specific packets from within the data store.

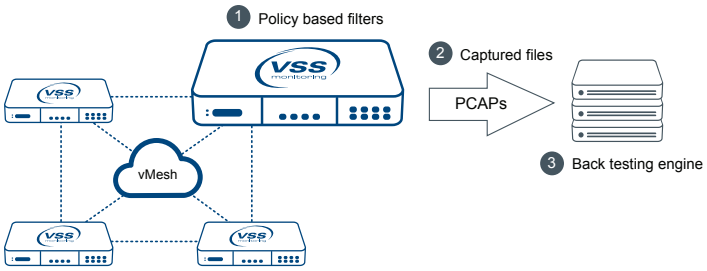
vIndex also supports the extraction of the above listed fields for packets inside GTP and GRE tunnels, which makes vIndex useful for cell phone service provider networks, where those protocols are widely used.

Example Use Cases with vSpool/vIndex and Related Planning Considerations

The following outlined use cases are examples of deployment scenarios in which either vSpool or vIndex can be used to enhance visibility for existing toolsets or to facilitate data delivery to storage platforms/big data systems.

PCAPs for Back Testing Trading Algorithms

Algorithmic trading is a type of securities trading in which the decisions to buy or sell securities are made by algorithms rather than by human traders. These algorithms use sophisticated technology to model the markets and use the market data to detect and exploit trading opportunities to make money.



It is estimated that 70% of the trading volumes on US exchanges are generated by algorithms trading securities, rather than humans. Examples in the past (Long Term Capital, Knight Capital) have shown what a bad algorithm can do to a company. Hence it is critical to test these algorithms thoroughly before letting them trade on live markets.

The market data feed, which is an input to these algorithms, is delivered from the exchanges in the form of a UDP flow. This UDP feed can be recorded in a PCAP format and then played back to an algorithm under test in order to observe its behavior. Thus, an algorithm could play back the prior day's PCAPs and trade on this historical information. The behavior of the algorithm can then be observed in order to test whether it would have made or lost money on the prior day if it were actually trading. Similarly, the PCAPs from exceptionally volatile days can also be used to test corner cases of the algorithm to make sure it doesn't fail in a catastrophic way.

Some points of consideration of such a use case would be:

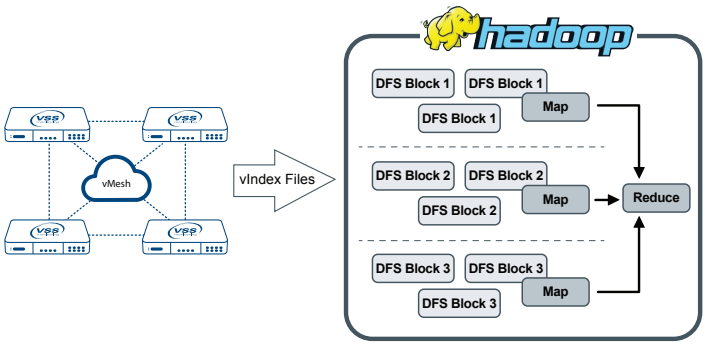
- Filtering – Market data feeds are sent on a specific UDP port. The feeds also tend to be sent redundantly on two separate UDP ports. The NPB layer can be used to filter out any non-feed traffic or the redundant port traffic.
- Storage Requirements – The amount of data stored depends on the volume of market activity on any given day. The total amount of storage also depends upon the number of days for which the data is stored in order to back test an algorithm
- IO Throughput – The data rate of the market feed will determine the IO throughput necessary. The data rate for equity market feeds tends to be much lower, for example, than those with options.
- Dropped Packets – In order to accurately simulate market data conditions, it is critical that the solution not drop any packets. In order to ensure this, the server should have enough processing power to receive and store the PCAP from the NPB layer. At a minimum, the server should have 1 GHz of processing power for every Gigabit of data received over a TCP connection. The NPB layer will also need to ensure

that microbursts of traffic do not result in packet loss for the captured data as well, by use of extended buffering.

- Latency between vSpool and FTP Server
- vSpool performs best when the latency between the vSpool module and FTP server is kept under 100microseconds RTD. Keeping latency under this value can help prevent performance degradation and packet loss.

Real Time Identification of Anomalous Flows using Metadata

As covered earlier in the vIndex section, vIndex files contain L2-L4 header information from each packet within a vSpool PCAP. As text files, they can be ingested and processed in a Hadoop cluster, for the performance of a wide range of analytics.



An example use case is identification of anomalous flows. Using Hadoop and the vIndex files, it is possible to classify the amount of data exchanged by different hosts within a network with external hosts. It is also possible to establish baselines for various protocols used on the network. For example, a baseline of 1% of network traffic may be established for DNS.

Having established these baselines characteristics, it is very simple to identify departures from these baselines to be identified for further analysis. For example, a sudden surge in DNS flows might indicate a DNS tunneling exploit in progress. Similarly, a sudden introduction of large FTP flows may indicate unusual activity.

Some issues to keep in mind when considering a use case of this type include:

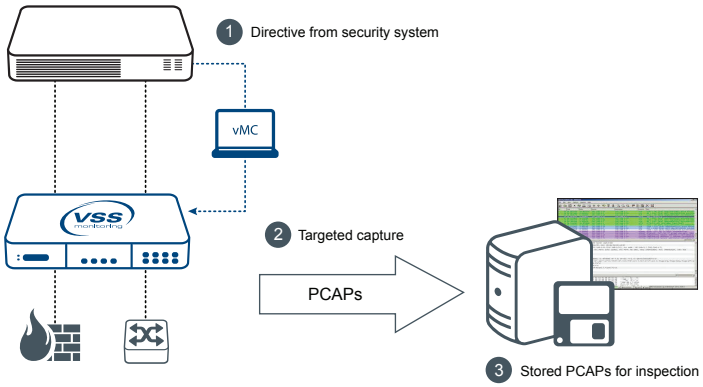
- **Identifying relevant fields** – vIndex extracts packet metadata prior to distribution to a storage platform or big data system; so it's important to identify all the fields that might be of interest when running analytics. While vIndex can extract all L2-L4 headers, each field can be enabled and disabled by the user. Reducing the number of fields extracted reduces the storage costs but also reduces the amount of information available for the analytics tools.
- **Staging Server** – vIndex (like vSpool) transfers files over FTP. Hadoop clusters don't natively accept FTP and hence a staging server is necessary to perform the protocol translation between FTP and the formats understood by Hadoop (JSON for example). Another option is to use a central storage solution, which accepts data over FTP and makes this data available to Hadoop clusters as needed.
- **Frequency of Batch Jobs** – Hadoop is primarily a batch processing technology. Thus, in order to schedule processing in near real time, batch jobs must be run frequently. How frequently these jobs are run determines the response time on anomalous flow detection.

Targeted PCAP Capture for Security Response

In this use case, vSpool is leveraged to provide "as needed" additional storage for an inline security tool based on the detection of anomalous traffic or other qualifying event.

When the inline security tool identifies a suspicious flow, it has the option of using an API call to the vMesh NPB layer to begin spooling that particular flow to a PCAP. Limiting PCAP generation/capture to only those that contain packets of interest is very useful in reducing storage requirements, and can make retrieval quicker.

A related use case would involve configuring vSpool to continuously create PCAPs of all flows seen on the network. Elements within the big data analysis structure can then start whitelisting flows that it deems safe, and adapt the VSS filters to match, which would prevent them from being archived. PCAP creation would be refined over time to be limited only to flows that are likely to be of interest in the future.



Both of these related use cases combine the power of inline security tools, vSpool, and the programmability of the vMesh NPB layer provided by the VSS APIs.

Some points worth considering when planning this solution in an enterprise are:

- **IO Throughput** – The FTP server needs to be able to keep up with the data rate being produced by vSpool to avoid holes in the captured data.
- **Latency between vSpool and FTP Server** – vSpool performs best when the round trip delay/latency between the FTP server and

vSpool is kept to a minimum. Thus, to avoid any packet loss it is best if the latency can be maintained at less than 100microseconds.

- **Processing power of the FTP Server** – Since the FTP server will receive large amounts of data from vSpool, it's necessary that it have the specifications to tolerate the volume. At a minimum, the server should have 1 GHz of processing power for every Gigabit of data received over a TCP connection.
- **Packet Optimization** – vSpool can be combined with Packet Optimization applications such as vSlice (conditional packet slicing) and filtering in order to refine the traffic to be captured, which minimizes the storage required for PCAPs.
- **Storage Requirements** – Storage requirements will be effected by the length of retention, whether the traffic is being pre-filtered by the NPB, and whether the white listing or black listing approach is used to identifying flows for capture.

Advantages of BDV Architecture

Bringing network data into an organization's big data platform has advantages on two fronts: (1) Cost savings on network analytics and forensics by augmenting their visibility or allowing data to be shared across multiple toolsets, and (2) the possibility of leveraging new analysis based on processing the data alongside other data sources (such as CRM).

For more information please contact us at info@vssmonitoring.com