Global Information Assurance Certification Paper

Implementing Full Packet Capture

GIAC (GCIA) Gold Certification

Author: Matt Koch, matt@altitudeinfosec.com
Advisor: Stephen Northcutt
Accepted: TBD

Abstract

Full Packet Capture (FPC) provides a network defender an after-the-fact investigative capability that other security tools cannot provide. Uses include capturing malware samples, network exploits and determining if data exfiltration has occurred. Full packet captures are a valuable troubleshooting tool for operations and security teams alike. Successful implementation requires an understanding of organization-specific requirements, capacity planning, and delivery of unaltered network traffic to the packet capture system.

1. Introduction

Deployment of a full packet capture system requires careful planning and an understanding of network architecture to successfully implement (Bollinger, Enright, & Valites, 2015). The information that a full packet capture system provides can be invaluable in case of a security incident and a source of frustration if incomplete (Sanders & Smith, 2014). Successful implementation relies on three key factors. First, planning for organization-specific requirements including minimum retention and where to capture network traffic. Second, delivering unaltered traffic to the packet capture system. Third, sizing the packet capture system to process and store the required network traffic.

1.1. The Need for Full Packet Capture

Full Packet Capture offers the virtual equivalent of a physical security camera monitoring the entrance and exit to a building: constantly recording (Sanders & Smith, 2014). Most network security tools rely on a negative security model: detecting known malicious traffic usually based on specific signatures. A negative security model is problematic in the event of zero-day exploits, new malware or attacks that simply do not have an existing signature (Vacca, 2014). Full packet capture enables a security analyst to review all of the system's communications which other security tools may not detect. Additionally, full packet capture allows for retrospection: replaying old traffic through new detection signatures. Retrospection can be used to determine if exploitation occurred before a detection signature or before a patch is released. The data gathered can also be used extract malware samples or write detection signatures (ISACA, 2013).

1.2. Regulatory Requirements

For most industry regulations, full packet capture is not explicitly required. There are however indirect requirements in several frameworks. For example NIST SP800-53 which requires logging of events "adequate to support after-the-fact investigations" of security incidents and "identifying the information involved" in the case of a security incident (NIST, 2016). The Health Insurance Portability and Accountability Act of 1996 (HIPAA) contains similar language: "Implement hardware, software, and/or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information". The additional guidance provided by Department of Health and Human Services does not explicitly mention any information security technology and leaves the organization to "determine reasonable and appropriate audit controls" (Department of Health and Human Services - USA, 2007).

2. Requirements

When planning a full packet capture deployment, several decisions are required that will affect capacity planning: Where to place full packet capture monitoring? What to monitor? What are the data retention requirements? Moreover, what redundancy or scaling requirements exist?

There are a variety of open source and commercial full packet capture solutions available. Some are purely software to be installed on a customer provided system; others offer a fully integrated hardware and software solution that is turn-key. Open source software lacks the standardized hardware and software support included in commercial solutions. The type of solution is organization-specific decision based on budget, procurement requirements, required features, labor hours available and existing organizational preference for open source or commercial solutions. Regardless of the type of full packet capture system chosen: where the system is monitoring and how much data the system will monitor determines the size and architecture required.

2.1. Placement of full packet capture system monitoring

In general, trust boundaries between a trusted and untrusted network are recommended for deploying full packet capture (Bollinger, Enright, & Valites, 2015). For example a connection from a corporate office containing employee workstations connecting to the internet. Other examples might include traffic from the untrusted Internet into a hosted web server network.

Using the corporate office example, Figure 1 shows the ideal network tap placement on the internal side before the outbound firewall. In environments using Network Address Translation (NAT), poor tap placement obscures network visibility as shown in Figure 2.

Microsoft Office User Microsoft Office User

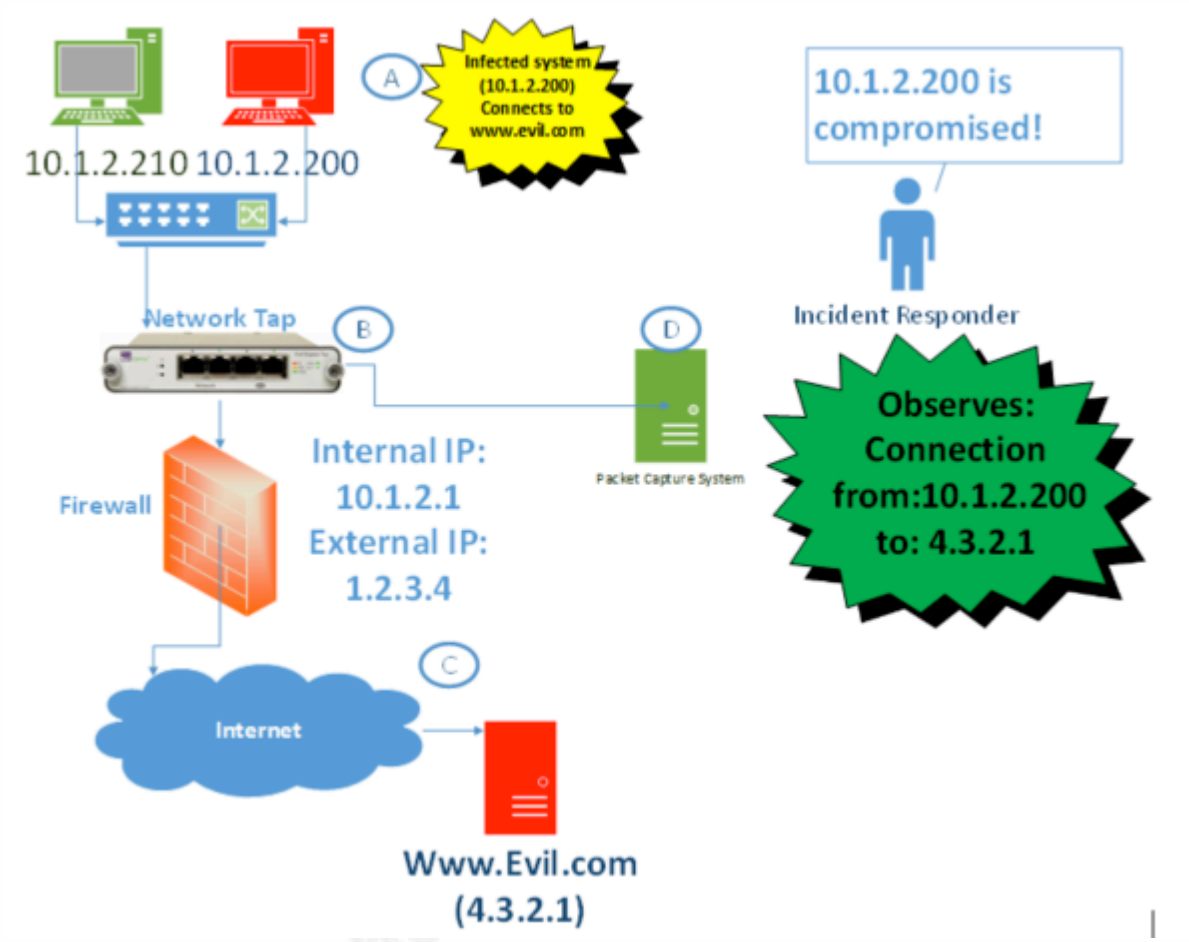

Figure 1: Ideal Tap location for monitoring end-user traffic: The incident responder can identify the compromised system.

Microsoft Office User Microsoft Office User

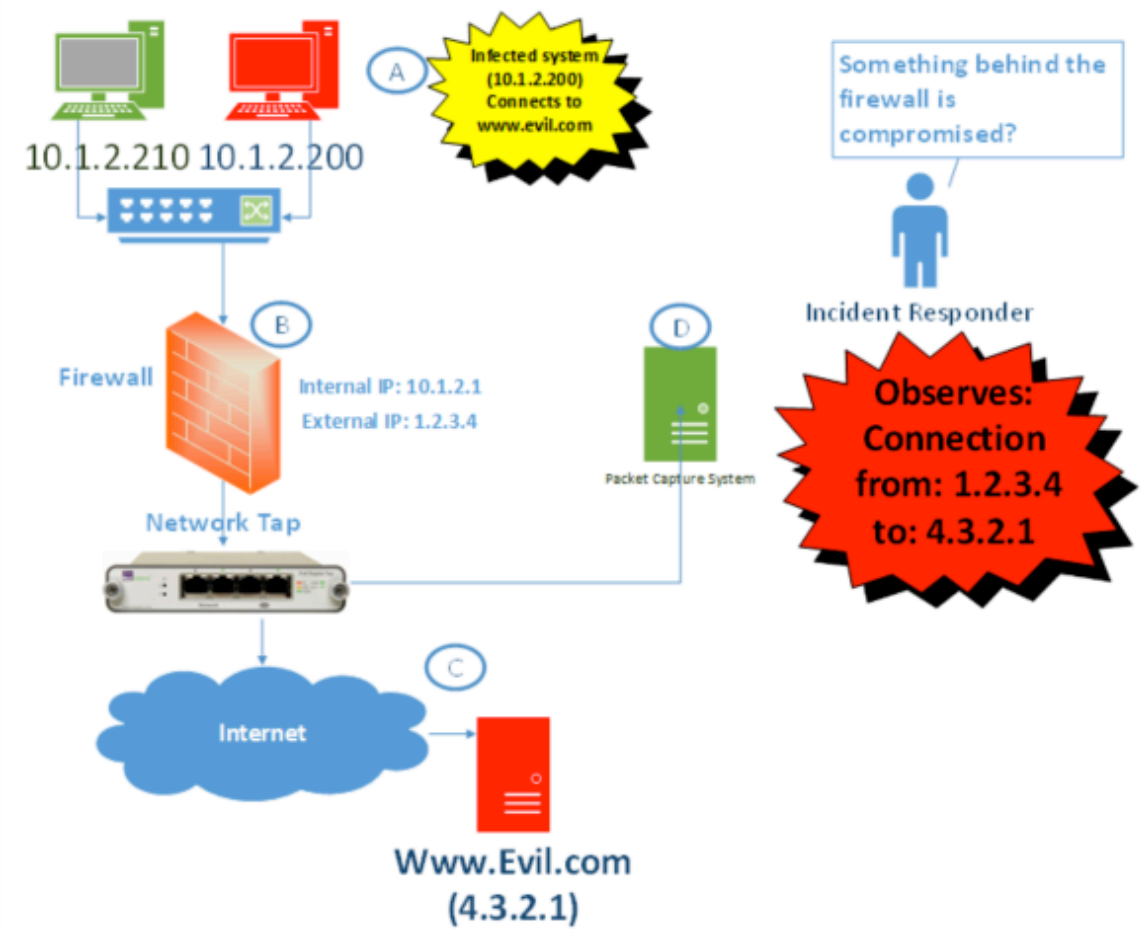

Figure 2: Poor Tap location for monitoring end-user traffic: The incident responder cannot identify the compromised system.

2.2. Storage Requirements

Is the organization required to retain data for a specific amount of time or dispose of data within a particular amount of time? Alternatively, what is the current average incident detection and response time? A 24/7 staffed security operations center may need a shorter timeframe than a security team that is only available during business hours. As a best practice, consider at minimum storing traffic from the previous night for review (Mowbray, 2008). If formal security incident metrics are available, use the mean time to detect (MTTD) to establish a minimum time requirement for storing packet data. Capturing MTTD can be a valuable metrics for describing how effective an organization is at spotting attacks. In the context of a full packet capture system: MTTD can provide a realistic minimum storage time requirement.

The unit of measure for network connection speeds is bits per second (bps): most commonly converted to Megabits (Mbps) and Gigabits (Gbps). Often overlooked is the duplex setting of the monitored connection. For Example, a 1 Gbps full duplex connection can send a maximum of 1 Gbps and receive 1 Gbps

Microsoft Office User Microsoft Office User

simultaneously (Spurgeon & Zimmerman). Using a full duplex 1 Gbps connection as an example, the packet capture system could be ingesting as much as 2 Gbps of traffic if both directions of the connection are fully utilized.

After understanding the speed of the connections monitored and retention time, the storage requirement needs to be converted from bits per second (a measure of connection speed) to a unit of capacity. The unit of measure for hard drives or other storage mediums is MegaBytes (MB), GigaBytes (GB) or TeraBytes (TB). Shown below, Figure 3 converts the connection speed, average utilization and required hours of recording to an estimated storage capacity requirement.

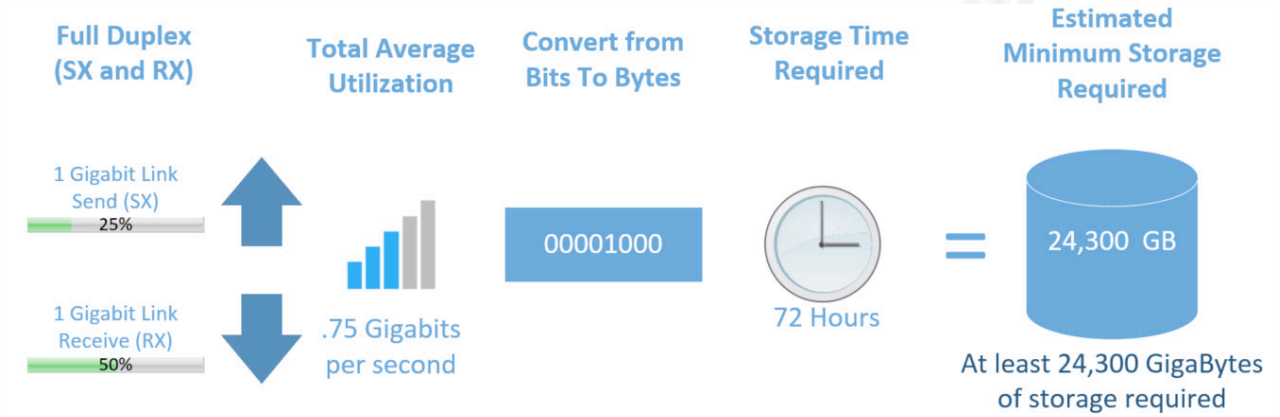

Figure 3: Calculating estimated minimum storage required for 72 hours of full packet capture on a 1 Gbps link with average 25% SX and 50% RX utilization.

Based on the incoming data rate the full packet capture system may need higher performance drives such as solid state drives (SSD) or Redundant Array of Independent Disk (RAID) controllers with write cache or other features to increase write performance. Volume calculations will be an important component. For example, if trying to use bare SATA 3 drives with a 6 Gigabits per second interface (Sandisk, 2016) is monitoring a 10 Gbps (1.25 GigaBytes per Second) connection, the hardware will not be physically capable of writing that data to disk in fast enough to keep up with storage requirements. When the 10Gbps connection nears full utilization packets will be buffered to memory and may not be written to disk if network utilization remains high or the buffer reaches capacity (Symantec, 2014).

3. Ingesting Traffic

The Full Packet Capture system needs to receive the original, unmodified network traffic. Several factors must be addressed to ensure the original traffic reached the full packet capture device. Many of these issues also apply to any other network-based security tools including intrusion detection and intrusion prevention systems. Ideally, the network traffic will be gathered with minimal impact on the network performance.

3.1. Tapping and Acquiring Packets

Physical network taps are the most preferred method of monitoring network traffic. Using a physical network tap ensures that the packets will not be modified in from the original. Many commercial network tap vendors are available including Ixia/Netoptics, Gigamon, and VSS Monitoring. When comparing tapping traffic to other methods such as a network SPAN (Switched Port Analyzer) switch port, there are several other advantages. Without the need to duplicate packets to a SPAN switch port, no additional load is placed on the switch. Many switch manufacturers also place higher preference on switching packets on SPAN interfaces and will routinely drop packets under even minor utilization to preserve the speed of the packet switching (Cisco Systems, 2016).

The packets observed on the network will differ from the packets shown by the switch SPAN (Cisco Systems, 2012). Due to the switch buffering the traffic and in some cases dropping malformed traffic. Given these factors, SPAN ports cannot provide a reliable source of traffic monitoring.

3.2. Physical Interfaces Capturing Packets

As discussed in the previous section, receiving the original, unmodified traffic as seen on the network can be important for accurately detecting attacks or troubleshooting issues on the network. Because the network interface receiving the traffic is not originating the traffic, there is no flow control on the amount or rate of the packets sent to the packet capture system. The lack of flow control requires a high-quality connection and a high-quality network interface card capable of receiving and processing traffic faster than a typical system. Popular manufacturers of these specialized capture cards include Endace, Napatech, CSPI, and Myricom. In large high speed networks, specialized packet capture cards should be considered (Sanders & Smith, 2014). These specialized cards claim zero packet loss and are highly specialized for network traffic monitoring using FPGA (Field-Programmable Gate Array) or ASIC (Application-Specific Integrated Circuit) technology (CSPi, 2016). Alternatively, for smaller networks with a small budget: high performance can be achieved using specialized software such as “PF_RING” while using supported commodity hardware (ntop, 2016).

Once the unmodified copy of the traffic is received by the full packet capture system via the network tap, the next place that the original packets could be modified is by the receiving network interface card. Network interface cards can include specialized chipsets to offload various network functions from the operating system or the CPU to be performed directly on the network interface card. Some examples include offloaded fragment reassembly, checksum validation and onboard buffering (Burks, 2011). Depending on the manufacturer, firmware, and operating system drivers the offloading sessions may be enabled by default. To avoid affecting the network traffic additional configuration is required to disable features that affect the incoming packet stream.

3.3. Packet Filtering

In some cases, not all network traffic requires monitoring or provides limited value (Bollinger, Enright, & Valites, 2015). Some examples might include scheduled high-volume backups, traffic to/from the full packet capture device itself or duplicative traffic monitored by another full packet capture system. Many packet capture tools allow the administrator to add these exceptions. Most commonly in Berkley Packet Filter (BPF) syntax. Additionally, BPF is used for packet filtering in many open source security tools including Snort, Suricata, Moloch and openFPC (AOL, 2016).

3.4. Packet Brokering

If the amount of traffic is larger than any single system can process, it is important to develop a scalable architecture. A 1:1 relationship of network tap to a single security monitoring tool is not scalable. In large, complex networks, multiple taps are required to achieve network visibility. Multiple network monitoring tools may need access to the same or a subset of monitored network traffic. Additionally, many organizations have availability and minimum redundancy requirements.

To address many of these requirements a class of devices usually referred to as "Network Packet Brokers" exist. Popular vendors include Ixia/NetOptics, VSS Monitoring, Gigamon and BigSwitch/BigTAP. Features vary between vendors but common functions include:

- **Load balancing:** spreading monitored traffic across several devices (sometimes referred to as "IDS load balancing").
- **Decoding and processing** of certain traffic: removing VLAN tags, inspecting traffic within MPLS or layer two routing protocols.
- **Filtering based on Layer 2-7:** for example send only HTTP/HTTPS traffic to a Network-based Web Application Firewall)
- **Decryption**: After uploading private encryption keys the packet broker can provide a decrypted traffic feed t o a device.

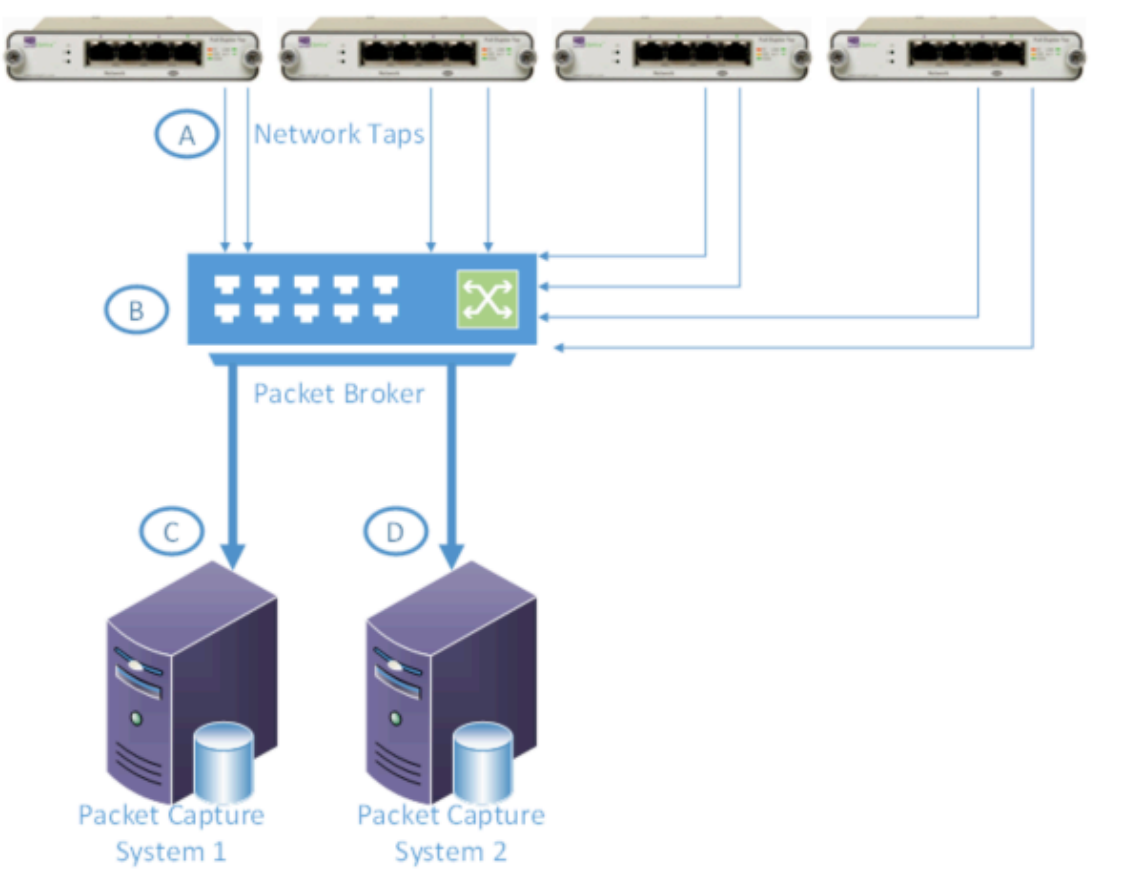

Figure 4: Packet Brokering Architecture

As shown in Figure 4, Network Taps (shown as "A") are connected to a Packet Broker device (Shown as "B"). The Packet Broker device ("B") received the traffic and based on its configuration outputs the traffic to two or more monitoring devices. Allowing for load balancing to a pool of two or more network monitoring devices (shown as "C" and "D")

If the traffic is incorrectly load balanced by sending 50% of the traffic to "C" and 50% of the traffic to "D": the single session is split amongst two devices (as shown in Figure 5). Analyzing the traffic reveals missing TCP segments using a tool like Wireshark (as shown in Figure 6).

The packet broker must inspect both the IP Header (including IP identification number and fragment number) and the TCP header (both the sequence and acknowledgment number) as part of its load balancing scheme. Using the IP and TCP header, the packet broker will ensure the entire conversation arrives at a single packet capture system (as shown in Figure 7).

Microsoft Office User Microsoft Office User

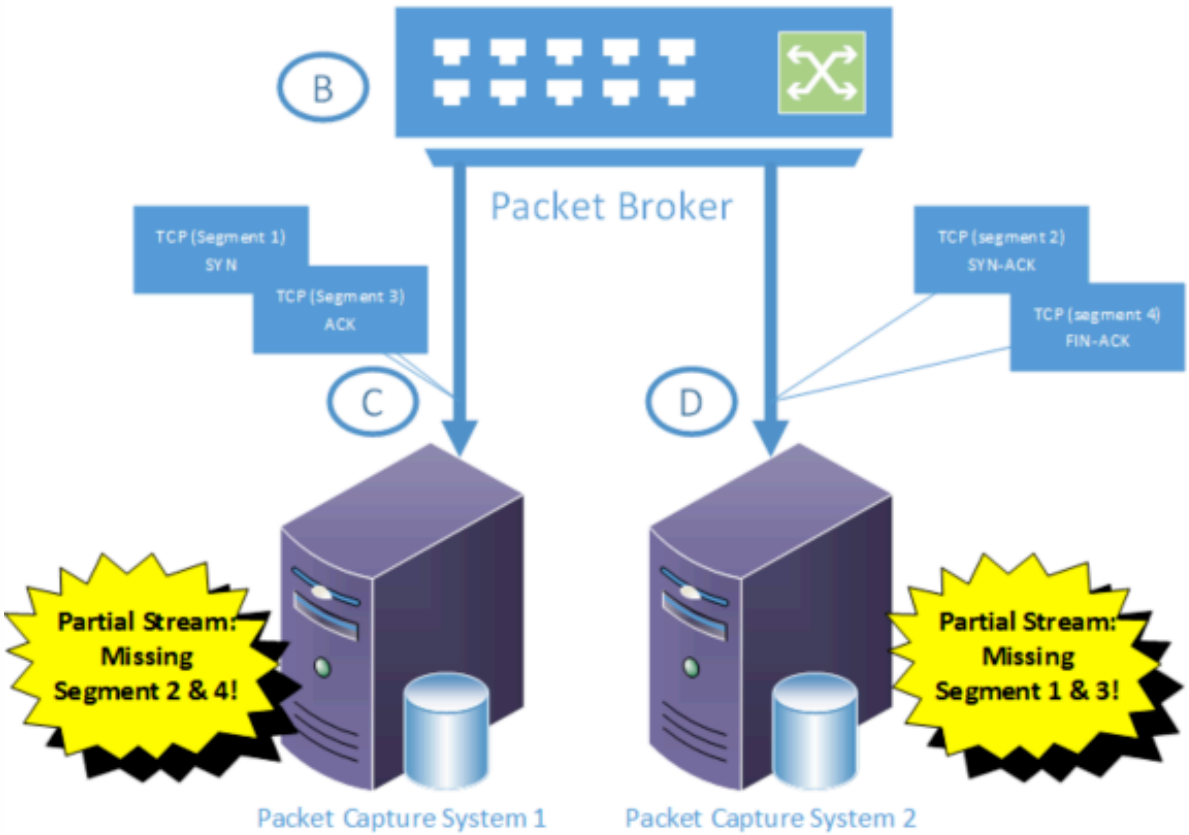

Figure 5: Packet Broker Breaking the TCP Session while load balancing between system C and System D.

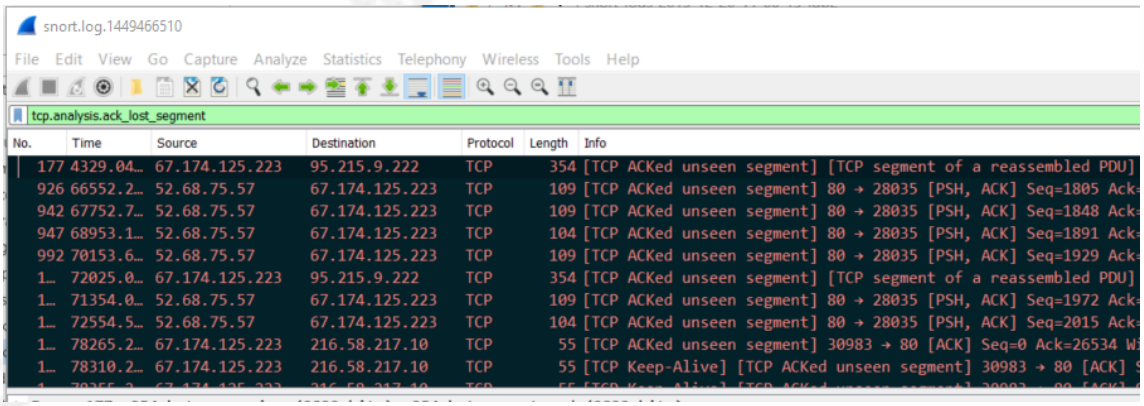

Figure 6: Missing TCP Segments, shown in Wireshark Analysis

Microsoft Office User Microsoft Office User

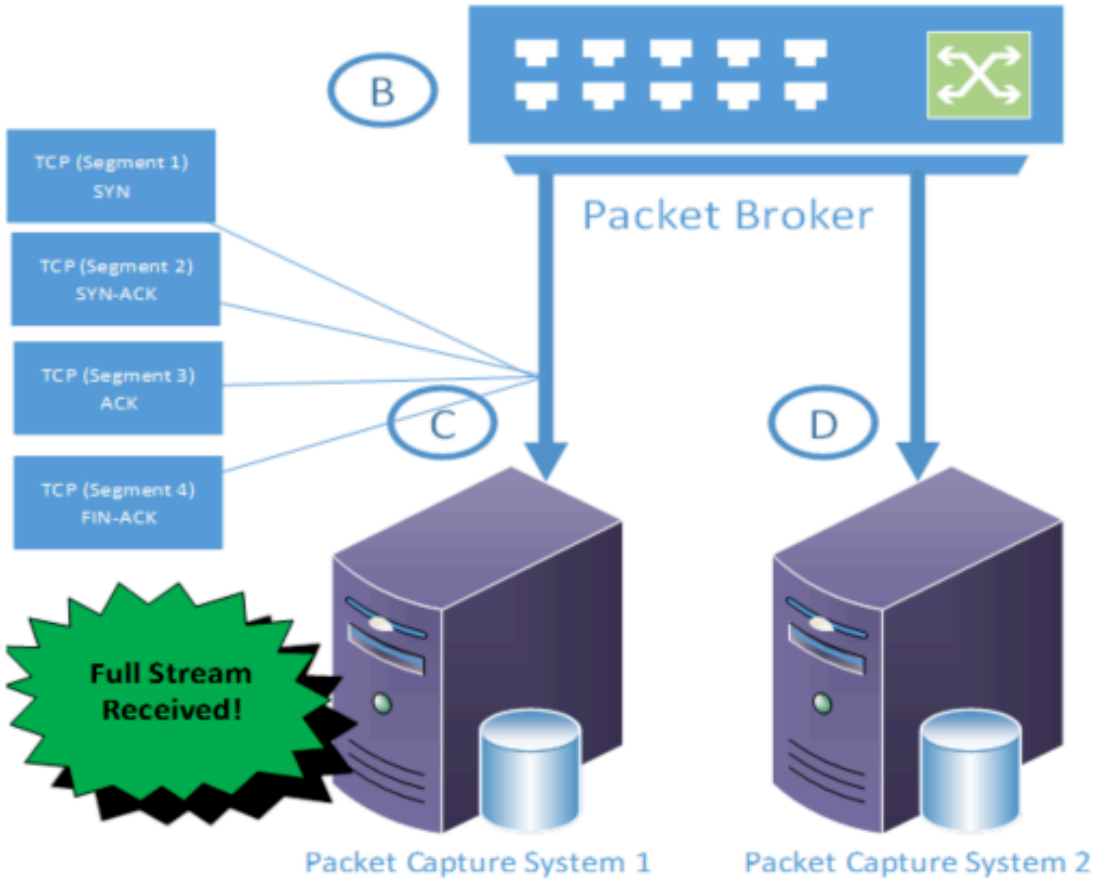

Figure 7: Packet Broker preserving the TCP Conversation while load balancing between system C and System D

3.5. Adding Metadata

Most raw network packets do not include any date or time information other than the relative sequence of the packets received. Because of this, full packet capture software must add date and time information to the stored packet data for later search. Accurate time information is critically important for legal proceedings or criminal investigations. Network Time Protocol (NTP) or in more critical environments a dedicated Global Positioning System (GPS) receiver can be used to keep the full packet capture system set to current time (Nucci & Papagiannaki, 2009). The time zone configured should be consistent with other security tools and when possible use coordinated universal time (UTC) to allow for easier data correlation amongst multiple security tools (Bollinger, Enright, & Valites, 2015).

Additional data enrichment can also be performed as the full packet capture systems processes incoming packets. Useful features include Geolocation to IP address information, found in Moloch and many other packet capture systems (AOL, 2016).

Microsoft Office User Microsoft Office User

Commercial solutions such as Protectwise offer IP reputation and intrusion analysis of captured network traffic in both near real-time and retroactively as new signatures are added and updated (Protectwise, 2016). The amount of metadata gathered must be included in storage calculation and capacity planning.

4. Storing Traffic

Once the Packet information is received, analyzed and enriched the appropriate metadata, it can be stored for later analysis and search. Having adequate system resources including CPU, memory, and high-speed storage is crucial to capturing the entirety of the packet stream.

4.1. Writing Packet Data to Disk

When the physical interface receives the traffic, the full packet capture software may apply filtering logic as discussed in 3.3 Packet Filtering. The packet capture software then reads the remaining packets from a memory buffer. The amount of incoming traffic, the speed write operations to the disk, processor speed and the amount of available memory for buffering can all affect the completeness of the packet capture data (Symantec, 2016). The packet data is them enriched with additional metadata and written to the disk in a file format used by the packet capture software. The amount of metadata gathered and the amount of indexing performed will also increase the amount of storage capacity required.

The amount of disk space available to the packet capture software may be a predefined value or a percentage of the total disk space. Moloch for example uses the "FreeSpaceG" setting which can specify size or percentage free disk required. As disk storage nears a preset capacity, the system must clear space. Deleting the oldest data first allows the current packet data to write to storage: known as a First-In-First-Out queue (FIFO).

4.2. Indexing and Searching Packet Data

The metadata described earlier needs to be stored and quickly searchable when an investigation of the packet data is required. An example might be searching the full packet capture system for a 10-minute window of time for a specific IP address of a compromised system. The software used to store the metadata and indexes differs depending on the solution used. For example, the open source packet capture software "Moloch" uses ElasticSearch to store and search packet data and additional metadata (ElasticSearch, 2016).

5. Testing and Monitoring a Full Packet Capture System

Testing and ongoing monitoring are required deploy a healthy packet capture system. Because of the nature of traffic monitoring, the connections to the monitoring interfaces are not flow controlled.

One method to check packet health is periodic monitoring of incomplete TCP traffic conversations. Observing TCP segments without corresponding reset (TCP RST or reset flag) or retransmits (a repeated packet with the same sequence number as the previous packet) indicate incomplete monitoring. Another example is missing TCP acknowledgment (SYN-ACK), but observing the same TCP session continuing (SYN or FIN-ACK). As shown in Figure 8, lost segments can be monitored using the TCP session analysis features of Wireshark or the text-based equivalent "tshark".

```
tshark –eth0 -R tcp.analysis.lost_segment || tcp.analysis.ack_lost_segment
Capturing on 'eth0'
 98 16.098870093 183.X.X.X -> 192.168.100.23 TCP 66 [TCP Previous segment
not captured] 30987 > EtherNet-IP-1 [FIN, ACK] Seq=1336 Ack=1852 Win=33024
Len=0 TSval=61084074 TSecr=3650428229
```

Figure 8: Using "tshark" filters to detect packet loss on the full packet capture system.

Checking for missing TCP segments without an observed retransmit is a tell-tale sign of packet loss between the source of the traffic and the full packet capture device. In many cases, these lost segments or errors will not be captured in the operating system interface counters ("netstat -su" or "netstat -st"). This packet loss could be due to over utilization of the packet capture system, a physical interface issue, issues with the network tap or SPAN port.

Network interface telemetry is also a valuable system health indicator and can monitor overall interface utilization and throughput as well as provide capacity planning data. Network interface statistics can be easily captured using SNMP (Simple Network Management Protocol) polling at regular intervals.

6. Conclusion

Full Packet Capture systems are a valuable tool for incident response. Deploying full packet capture systems requires careful planning and an understanding of the organization's network. Equally important is continuing to monitor that the packet capture system operates as expected in the event of an incident.

7. References

AOL. (2016, March). *Settings · aol/moloch Wiki · GitHub*. Retrieved from GitHub.com: https://github.com/aol/moloch/wiki/Settings

Bollinger, J., Enright, B., & Valites, M. (2015). *Crafting the infosec Playbook.* O'Reilly Media, Inc.

Bollinger, J., Enright, B., & Valites, M. (2015). *Crafting the InfoSec Playbook: Security Monitoring and Incident Response.*

Burks, D. (2011). *When is full packet capture NOT full packet capture?* Retrieved from Securityonion.net: http://blog.securityonion.net/2011/10/when-is-full-packet-capture-not-full.html

Cisco Systems. (2012). *Catalyst Switched Port Analyzer (SPAN) Configuration Example*. Retrieved from Cisco.com: http://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/10570-41.html#anc67

Cisco Systems. (2016). *Catalyst 6500/6000 Switch High CPU Utilization.* Retrieved from Cisco.com: http://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/63992-6k-high-cpu.html

Cisco Systems. (2016). *Cisco APIC Troubleshooting Guide.* Retrieved from Cisco.com: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/troubleshooting/b_APIC_Troubleshooting.pdf

Department of Health and Human Services - USA. (2007, March). *HIPAA Security Series: 4 Security Standards: Technical Safeguards.* Retrieved from HHS.Gov: http://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/securityrule/techsafeguards.pdf

Department of Health and Human Services - USA. (2007, March). *HIPAA Security Series: 5 Security Standards: Organizational, Policies.* Retrieved from HHS.Gov: http://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/securityrule/pprequirements.pdf

ElasticSearch. (2016). *Moloch - How Elasticsearch is Powering Network Forensics at AOL*. Retrieved from Elastic.co: https://www.elastic.co/videos/moloch-elasticsearch-powering-network-forensics-aol

Gordon, & Hernandez, S. (2016). *The Official (ISC)2 Guide to the SSCP CBK.*

ISACA. (2013). *Responding to Targeted Cyber Attacks.*

Ixia. (2015, July). *What is a Tap, and Why Are Taps Critical to Network Visibility and.* Retrieved from Ixia: https://www.ixiacom.com/sites/default/files/resources/solution-brief/915-6855-01-inlinetapsvsspanports.pdf

Mowbray, T. (2008). *Cybersecurity: Managing Systems, Conducting Testing, and Investigating Intrusions.* Wiley.

NIST. (2016). *NIST Special Publication 800-53 Rev4.* Retrieved from NIST.gov: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

Microsoft Office User Microsoft Office User

Nucci, A., & Papagiannaki, K. (2009). *Design, Measurement and Management of Large-Scale IP Networks.*

Sanders, C., & Smith, J. (2014). *Applied Network Security Monitoring.*

Sandisk. (2016, May). *Difference between SATA I, SATA II and SATA III*. Retrieved from SanDisk Global Customer Care: http://kb.sandisk.com/app/answers/detail/a_id/8142/~/difference-between-sata-i,-sata-ii-and-sata-iii

Spurgeon, C., & Zimmerman, J. (2014). *Ethernet: The Definitive Guide 2nd Edition.* O'Reilly.

Symantec. (2014). *Packet Capture Buffer and Memory Usage*. Retrieved from Symantec.com: https://support.symantec.com/en_US/article.TECH221445.html

Vacca, J. (2014). *Network And System Security: Second Edition.* Syngress.

Microsoft Office User Microsoft Office User