

Traffic Monitoring with Packet-Based Sampling for Defense against Security Threats

Joseph Reves and Sonia Panchen

This paper describes the technology and large-scale deployment and use of a distributed network traffic monitoring system based on a packet-based sampling technology. It gives examples of various techniques making use of the resulting network traffic data to address network security issues.

I. INTRODUCTION

Network service providers are being faced with increasing disruption to network services because of a variety of security threats and malicious network service misuse. Such threats may originate externally or internally, and may occur at any time. To detect and respond promptly to this situation requires broad and continuous surveillance of network activity that provides timely and detailed information.

This paper describes the technology and large-scale deployment and use of a distributed network traffic monitoring system based on a packet-based sampling technology. It gives examples of various techniques making use of the resulting network traffic data to address network security issues.

II. PACKET-BASED SAMPLING MONITORING SYSTEMS

A. Rationale

Use or misuse of a network will generate network traffic, therefore monitoring network traffic can be an effective mechanism to identify, diagnose, and determine controls for network misuse.

Traditionally, network traffic monitoring has been achieved using probes. This has worked very effectively in shared networks where a single instrument can monitor all the traffic. However, with the trend towards switched, point-to-point networks, every port on a switch would have to be monitored to achieve the same visibility to network traffic. In addition, switches and routers make packet-forwarding decisions that affect the flow of traffic through a network. Understanding these traffic flows is critical to maintaining visibility to network use and misuse. Implementing traffic monitoring within a switch or router is an effective way to see traffic on all ports and the flow of traffic. However,

market trends have led to increasing bandwidth and decreasing switch and router costs, resulting in the requirement that an embedded monitoring system has little or no impact on switch or router cost or performance, especially since monitoring is secondary to the primary forwarding function of the device. It is possible to implement a traffic monitoring system, based on packet sampling, which meets these network equipment vendor requirements and also meets the end-user requirements for a monitoring system that enables prompt detection and responses to security threats:

- Network-wide, continuous surveillance since attacks can emerge from any point at any time.
- Data must be available in a timely manner, especially during network overload, to enable prompt response.
- Data must have sufficient detail to fully characterize a threat so that appropriate controls can be determined.
- The monitoring system itself must not make the network vulnerable to attack (i.e. the monitoring mechanisms must not impact performance during an attack)

B. Embedded packet based sampling for traffic monitoring

The multi-vendor technology, sFlow (RFC 3176 [9]), is a packet-based sampling technology that was designed to meet these criteria. sFlow consists of a packet sampling algorithm, typically performed by the switching/routing ASICs, and an sFlow Agent which is a software process that runs as part of the network management software within the device. The sFlow Agent combines flow samples (generated by the packet sampling function), interface counters, and the state of the forwarding/routing table entries associated with each sampled packet, into an sFlow datagram which is immediately forwarded to a central sFlow collector. This means that the sFlow Agent does very little processing of the data and minimizes the CPU and memory requirements. Meanwhile, a central sFlow collector receives a continuous stream of sFlow datagrams from across the entire network and analyzes them to form a rich, real-time view of L2-L7 traffic flows across the entire network. This system has the following advantages:

1) sFlow agent

Low cost - sFlow has minimal requirements on device resources since the sampling algorithm is simple and

performed in hardware and very little processing of sampled data is performed on the device.

No impact to device performance - the sampling algorithm is simple and performed in hardware. Under heavy loads and at gigabit speeds, the flow measurements will not be affected by device performance or be "clipped". This means that monitoring can be enabled continuously on all ports to provide network-wide surveillance.

Minimal network impact - the sFlow datagram uses efficient encoding of the data. A continuous, low volume, asynchronous traffic stream is much less taxing to the network than bursty data transfer. A typical implementation will result in 0.1% bandwidth utilization. This also means that the monitoring system does not expose a vulnerability to disrupt network service.

Robust - the sFlow system is designed so that the accuracy of any measurement can be determined. If measurements are lost in the transfer from agent to collector, the result is a slight decrease in the effective sampling rate.

Simple to implement - both the hardware sampling algorithm and the agent functionality are very simple. These qualities encourage pervasive implementation much in the same way that SNMPv1 did.

2) sFlow collector

Network-wide, continuous view - because sFlow Agents generate a constant, low volume of data, and the sampling function reduces the size of the data set that a collector manipulates, a single sFlow Collector can monitor and present a consolidated view of a network of thousands of switches.

Timely data available - because the sFlow Agents immediately forward data, the sFlow Collector can present an up to the minute picture of network traffic.

Detail - since analysis is performed at the sFlow Collector, detailed data is preserved. This means that full decoding of the sampled packets can be performed to generate L2-L7 traffic matrices and stateless signature characterization.

Technology based on this system has been deployed in a number of large enterprises and service providers. The following case studies on dealing with security threats are taken from experience with these deployments.

C. Cases studies

The following specific data network security applications are drawn from two environments.

In one environment, packet-based sampling technology has been pervasively deployed within an enterprise network and used for nearly a decade. In this environment the benefits are well understood, and the existing network infrastructure is deployed to support pervasive continuous sampling throughout the environment.

In the other case, sFlow instrumentation was introduced to provide a packet-based sampling capability within the core of an Internet Service Provider network.

III. CASE STUDY ONE - GRADUAL SEPARATION AND CONSTRUCTION OF A FIREWALL FOR A LARGE SCALE DIVESTITURE

The first case study is drawn from the experience of a large divestiture of globally distributed business units from a large multi-national company.

In 1999, Agilent Technologies was created as a "spin-off" of several businesses from Hewlett-Packard Company. While the necessary financial transactions to create Agilent as an independent company were completed relatively quickly, the separation of IT services – and in particular, the enterprise network infrastructure – took an additional two years of work to complete. During this time, Service Level Agreements for IT services governed the connectivity permitted between the intranets of each company.

An internal firewall between the two companies was constructed, and traffic flowing between the two enterprises was characterized in order to generate firewall rules that would gradually restrict non-permitted traffic as various service level agreements expired. This traffic characterization also supported a continuous "audit" of applications traffic that was actually flowing between HP and Agilent businesses.

In this case study, we will examine how packet-based sampling technology was used to create enterprise application traffic studies to support the configuration of this inter-enterprise firewall, and to audit the firewall performance over time. We will examine the evolution of the firewall from fairly gross, coarse-grained controls to fine-grained host/application level controls.

A. The Challenges: Large scale corporate divestiture

When the decision to divest the group of businesses that were to become Agilent Technologies was initially taken, the combined global corporate network consisted of approximately 400,000 IP hosts connected through a highly meshed wide area network spanning the Americas, Europe, and Asia/Pacific regions. Within business units, operations were often geographically dispersed. Many business applications systems were distributed, although many were concentrated in relatively few major data center sites serving both businesses that would become Agilent and businesses that would remain HP.

At the time of the split, it was estimated that over 1100 discrete business applications would need to separate, using a "clone and go" strategy. This was only the tip of the iceberg. Many other network based dependencies existed: network services infrastructure (proxy infrastructure, mail servers, socks servers, network time, name services, network news, domain authentication systems, and so on), as well as the kind of peer-to-peer collaboration that's common within a corporate network. File shares mounted between users, web based trouble tracking systems or

project status web sites – even the web site used to coordinate the network team’s split activities.

In short, the network conversation matrix between geographically and organizationally dispersed users, and the systems that supported them was seemingly impossible to visualize, much less unravel.

1) Connectivity during the split

The basic strategy to split the company called for Agilent to be completely renumbered out of HP’s net 15-address space, into new multiple class B networks. In practical terms, this meant creating a new company backbone infrastructure, while identifying subnets within the legacy net 15 network that were destined to be renumbered into the new address space. This made conversation analysis particularly difficult, as each and every subnet within the legacy network had to be identified as either “staying with HP”, “shared infrastructure”, or “destined for Agilent” in order to meaningfully analyze conversations traffic. During this phase of the split, the lesson we learned was that it was absolutely critical to have a systematic approach to associate your address allocations with business units. We had previously considered the need to analyze “inside/outside” traffic conversations primarily, but intra-business traffic analysis has ramifications for both acquisitions and divestitures.

Eventually, as the network backbone was created, five “enterprise” inter-company links were created between the new Agilent network and HP’s legacy network. As Agilent businesses began to migrate to the new network, conversation analysis between the two companies became much simpler to base on source and destination IP addresses. During this phase, communication between the two companies was not filtered or blocked in any way.

2) The challenges of instrumentation

In the early phase of the network split, HP was fortunate to have deployed an infrastructure that had included embedded packet-based sampling technology. This technology was originally developed at HP, and was implemented at the ASIC level in most of HP’s switching products – which, of course, made up the bulk of the layer two infrastructure in HP’s network.

InMon Corp. and HP agreed, that InMon could make use of this data, and provide scaleable, enterprise-class analysis applications. InMon’s applications provided the basis for the analysis required to complete the HP / Agilent network split.

Additional challenges in instrumentation included the need to gain visibility of the wide-area network links; all the existing instrumentation was based in layer-two switching equipment. In terms of scale, inter-enterprise traffic had to be collected and consolidated from five locations around the world, and integrated and presented at the inter-company level. In some cases we needed near real-time

views of this information, as we made changes to the access rules.

While the high-speed inter-company links were eventually instrumented as “choke-points”, we found that distributed, pervasive, embedded packet-sampling instrumentation gave us a highly accurate visualization of the traffic, with detailed drill-down information on the character of conversations within individual source and destination subnets.

3) Shared Services, phased connectivity

We should probably not have been surprised to learn that the majority of traffic, by volume, was composed of shared network services. Proxy traffic through caching proxy servers, SOCKS traffic, and DNS traffic were the greatest contributors to the traffic mix. Pulling out shared infrastructure services and considering them separately from other application traffic immediately clarified the requirement to duplicate network services, public internet access, proxy and mail infrastructure, and so on. As individual client configuration was difficult to manage broadly – we had a mix of HP-UX and Microsoft Windows clients – we eventually wound up having to disallow access by service for clients in Agilent and responding to reactive trouble calls from those desktop users that had not gotten the word to migrate their configuration.

B. Building the firewall

The entire process of network separation between the companies can broadly be divided into three major phases:

- Preparation
- Separation
- Diminishment

We will examine the role of traffic characterization in each phase.

1) Preparation

The preparation phase began shortly after the split was announced, and the realization dawned that literally hundreds of application systems – many of them home-grown – would need to be replicated for Agilent businesses.

From a network infrastructure perspective, this phase involved identifying IP subnets associated with business units that would certainly go with the new company, versus certainly staying with HP. This left a large, but manageable number of subnets to be classified and a growing number of subnets identified as “shared”. At the beginning of the split activity, we started with about 8100 individual subnets.

This phase was dubbed the “Mercedes” phase, reminiscent of the well-known automobile ornament with three sections. The strategy was to complete the classification of unknown subnets into one of the three dispositions – HP, Agilent, or Shared – and then to use traffic characterization in the shared subnets to determine the mix of “HP” client traffic versus “Agilent” client traffic.

Analysis tools

We used two primary network-based analysis tools during this period. The first was based on a collection of subnet population data from distributed Openview Network Node Manager machines, and provided an accurate near real-time count of the number of active IP nodes in each subnet. This offered a clear measure of progress both globally – as subnet dispositions were characterized – and locally, within each subnet. Total “shared” node counts gave us a picture of the progress of the “clone and go” activities.

The second tool was the off-the-shelf capabilities of the InMon traffic analysis applications, which offered a breakdown of application traffic in the shared server subnets to help us determine the mix of Agilent-bound clients versus traffic from clients remaining with HP.

“Mercedes” takes shape

At this point, the Agilent backbone network was still not up and running and systems were staged for migration in Net 15 subnets with an “Agilent” disposition. The plan at this point was to begin to firewall access between HP and Agilent subnets, but this was not possible until migration into Agilent address space could begin. The build out of the global Agilent backbone took somewhat longer than originally anticipated, due to delays with our network circuit providers.

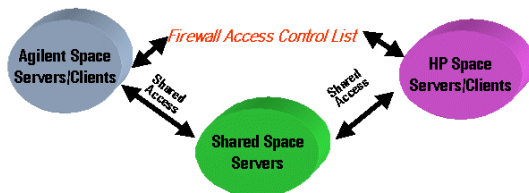


Figure 1 “Mercedes” Model

Shared subnets were gradually diminished as systems were cloned into new Agilent-bound servers and renumbered into Agilent subnets to stage for migration. The shared subnets were never completely emptied, as systems that would eventually host applications for Agilent under a trade customer agreement with HP remained.

2) Separation

The separation phase marked the beginning of migration from HP’s Net 15 address space into newly created Agilent address space. Network connections were established between Agilent’s new backbone and HP’s intranet at five major inter-enterprise locations. Each of these locations was a shared HP/Agilent site – and so the network infrastructure to connect the two enterprise networks was built around high-speed local area network connectivity. Practically, this meant that the switch fabric around the connections could be instrumented for traffic visibility, rather than enabling a router-based instrumentation method like cisco’s NetFlow.

Examining the traffic flow between the two address spaces during this migration process provided both a sense of the progress of the migration, as well as a notion of what types – and how much – client access from Agilent flowed back into HP’s network.

Scalability versus detail

Once again, the issue of scalability arose as we began to examine the traffic conversation detail across the two address spaces. As the goal was to create access controls that would eventually limit access from any Agilent source to specific HP destinations, we quickly focused on a “Destinations Report” representing traffic from Agilent source systems to HP destination subnets. Initial conversation detail analysis indicated that Agilent systems communicated with between 12,000 and 15,000 individual destination systems in HP’s network at the beginning of this phase. Clearly, we could not formulate decisions or take actions from this data without further analysis.

Analysis tools

Our primary tools in this phase utilized conversation and protocol information extracted from five InMon traffic server machines around the world, and consolidated and presented with custom-written perl scripts. The InMon server supports a simple scheme of data extraction based upon a query formulated in an http URL. Data can be returned as formatted HTML, or simple csv. Perl HTTP query modules were used to drive the extraction scripts, and interactive drill-down functions were implemented with CGI scripts. As is often the case, simple tabular presentation formats proved to be the most useful.

The analysis scripts were broken down into several components:

- Data extraction, and global consolidation
- A high-level “Destination Subnets” report, which could be sorted several ways
- Drill-down subnet conversation detail
- A drill-down report for Data Center Subnets, identifying services offered by server
- A real-time visualization panel for viewing all traffic from all five servers

The initial high-level Destinations Report (Figure 2 Destinations Report) presented a table of HP destination subnets, sorted by traffic volume and identifying Data Center Subnets. Note that each destination also indicates from which traffic server the largest volume of data is visible.

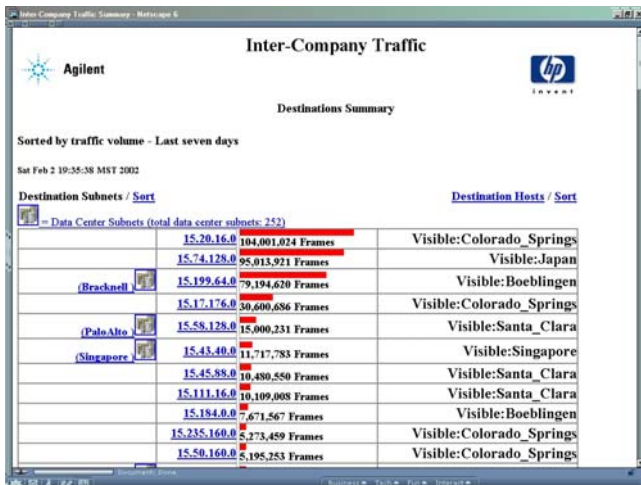


Figure 2 Destinations Report

Focusing on Data Center traffic in this phase helped to identify remaining servers to be either cloned, or migrated to long-term trade customer services. Simple java applet charts (Figure 3 Data Center Traffic Overview) provided a perspective on progress for each of the Data Center sites.

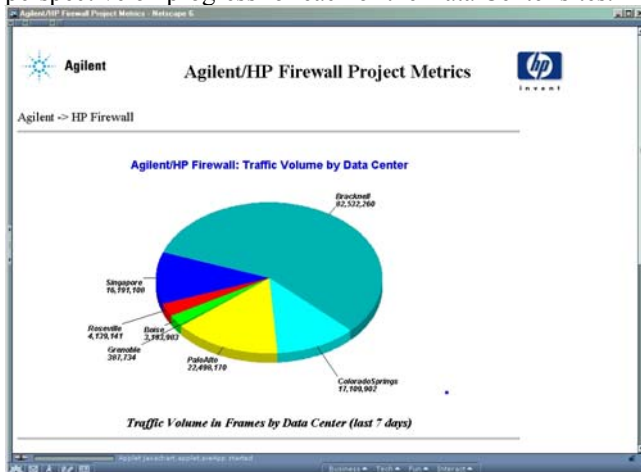


Figure 3 Data Center Traffic Overview

A divestiture within a divestiture

During this phase of the HP / Agilent split, an additional complication was introduced with the divestiture of an Agilent business unit. This created the requirement to use all of these traffic characterization processes again, in a more focused setting to identify shared services and network based dependencies for this business unit. It also accelerated the requirement to place network access controls between the business unit and the remainder of HP’s network in advance of the planned access controls for Agilent.

We used the same data collection and analysis methodologies with this smaller divestiture as we used to analyze the HP / Agilent split. Aside from the scalability issues being much simpler in this setting, the process worked in exactly the same way.

3) Diminishment

In the diminishment phase, we finally began to place access controls between the inter-company network connections to progressive diminish casual traffic. These access controls were constructed as cisco router ACLs on the inter-company routers. HP used a new compartmentalized access controls architecture to logically define all of the Agilent interconnect links as a part of the same security compartment, to coordinate global access controls consistently.

The initial strategy called for access controls to begin with the “observed” traffic from the consolidated traffic matrix data, and then progressively eliminate classes of access after appropriate communication. Network infrastructure services, for example, were among the first class of traffic to be eliminated.

This initial construction of the access lists was challenging, in that the discovered traffic matrix still showed communication to over 10,000 destinations within HP. Obviously, it was not practical to enumerate these destinations discreetly in an enterprise-class router access list. We began with a subnet destination level of granularity for non-data center subnets, and an individual host level of granularity for hosts within a data center subnet. This allowed us to examine whole non-data center subnets by traffic type, while focusing on data center services provided by individual hosts for migration to trade customer facilities.

Even with this multi-tiered granularity in our access lists, the conventional definition using explicit permit statements would have run somewhere between 70,000 and 130,000 lines in length. An additional innovation in the design of the access lists defined “anti-Agilent” address space, which shortened the definition to fewer than 3000 lines.

Analysis tools

As we implemented these controls, we needed a near real-time visualization of the traffic flow globally across the inter-company links. Our reporting up to this point had gathered and consolidated InMon traffic server data before analysis. Our requirement now was to implement the ACLs, and then watch the results live from servers all around the world.

We used a “panel” of charts (Figure 4 Real Time Visualization Charts) running from each of the servers to visualize the traffic at a high level, tuned to present the traffic class that we were working with.

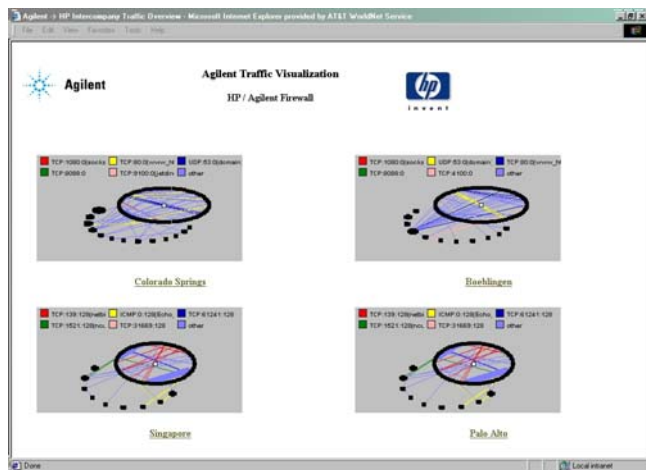


Figure 4 Real Time Visualization Charts

We constructed this panel with a simple frameset of html, embedding standard “Circles” format InMon applets into a table. As we implemented firewall access control changes, this visualization provided a quick check to insure that the traffic class we filtered was in fact blocked.

C. Ongoing analysis – “Smoke Test” for the firewall

On an ongoing basis, these traffic visualization and reporting techniques can serve to validate the functionality of access controls implemented at the firewall. Examining the traffic matrix collected from pervasively deployed instrumentation can reveal “leakage” in the performance of a firewall, as well as the existence of unauthorized “home-grown” packet-forwarding configurations implemented – deliberately or inadvertently – by the user community.

In a VLAN environment, this kind of analysis can reveal leakage between VLANs caused by vendor defects or mis-configuration.

This independent examination of the traffic matrix serves as a kind of “Smoke Test” for the performance of firewall configurations. Most of the firewall audit capabilities currently on the market are designed to either attack a firewall, or parse a firewall configuration file for consistency. Examining the traffic present on either side of a firewall provides an independent confirmation that the firewall is functioning correctly.

Hewlett-Packard and Agilent will no doubt continue to use these techniques for future business divestitures, acquisitions, and mergers.

D. Futures – Anomaly Analysis

Future applications of traffic analysis currently being investigated focus on anomaly detection and the identification and containment of malicious traffic streams.

With a pervasive packet-sampling capability deployed in the enterprise, it’s possible to examine traffic on a very broad scale in a very scaleable fashion. There are two areas of focus at the moment.

1) Conversation information

Anomaly detection in conversation information requires profiling “normal” conversations for either individual servers, or for areas of the topology. For example, it might be possible to profile “normal” traffic to Research and Development Labs, and examine anomalies that might represent unauthorized transfer of intellectual property. Examining the “fan-out” of conversations between clients and servers might help identify unusually broad communications from an individual client to many machines – a signature, perhaps, of an infection vector for a network virus.

2) Sampled packet stream inspection

Another opportunity to identify malicious traffic exists in the examination of the stream of sampled packets as they arrive. Typically, signatures that could be examined rely upon anomalies in the packet header, or content near the beginning of the packet. Signatures that could be examined would have to rely upon indicators present in an individual packet, as sampled collection would not provide for stateful connection inspection.

One advantage to sampled collection, however, is that it scales for pervasive deployment. This addresses the issue with state of the art Intrusion Detection Systems that promiscuously examine all packets, but only operate in a narrow scope of deployment before scalability becomes impractical. Pervasive sampled collection throughout an infrastructure does not require the identification of “chokepoints” for collection and analysis, and provides for multiple sample collection points in the path for any individual conversation. This is work for future study.

IV. CASE STUDY TWO – AN ISP DEFENDS AGAINST EXTERNAL SECURITY THREATS

The second case study is drawn from the experiences at a large ISP, using packet-based sampling technology to detect and respond to various security threats, including DoS.

This ISP is one of the largest Web hosting operations and is a leading provider of comprehensive Internet services. They provide services to customers in more than 170 countries. These operations are supported by a global infrastructure and systems including a Tier One network.

A. The challenges – Maintaining service availability

This ISP’s business model is dependent on the availability of services that it can provide. This affects their ability to meet SLAs and their ability to attract new customers because of the high quality of service that they can provide.

The frequency of the attacks directed at this ISP is consistent with the observations reported by Moore, Savage, Voelker [9] These attacks present a major challenge. They flood network traffic at a particular target, often from a variety of topologically distributed sources. The resulting congestion at the target and of network

resources can lead to decreased availability of hosted services and consequently potential loss in revenue.

This ISP also uses traffic monitoring to help its customers identify and diagnose other security breaches, for example whether hosts have been compromised or experienced unauthorized access.

Because attacks can come from a variety of unexpected sources, effective detection requires continuous, network-wide surveillance. This surveillance should not impact network performance or generate large quantities of data that make real-time analysis intractable.

The approach originally used by this ISP to identify security threats was to monitor the interface counters at the edge. When the counters spiked, they would manually work back down the tree of switches looking at interface counters, looking for the highest to identify which interfaces were affected. This was a time consuming process and did not give IP information. To improve on this situation, the ISP implemented a packet-based sampling, network traffic monitoring system, sFlow.

1) Monitoring system configuration

Since the main concern of this ISP is externally sourced attacks. The primary focus was on instrumenting the up links of the core switches so as to give visibility to ingress and egress traffic. The second priority was monitoring internal site traffic, using embedded sFlow in the distribution switches.

This configuration (Figure 5 ISP Network and Monitoring System Configuration) gives real-time, media-speed L2-L7 traffic flow visibility in and out of the site, and within the hosting site.

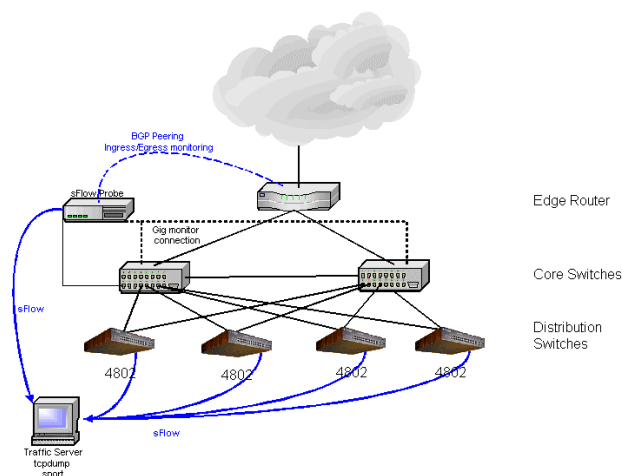


Figure 5 ISP Network and Monitoring System Configuration

The following sections describe the techniques, using sFlow, that have been used to characterize and take action against security breaches.

2) Techniques using sFlow data to identify security breaches

There are a few basic techniques that this ISP has used to identify various types of security threats – DoS, intrusions, and compromised hosts:

- From continuous monitoring of traffic patterns, establish baselines for normal network usage. For example typical link utilization, service usage.
- Raise events when traffic patterns deviate from established baselines.
- Look for the Top N hosts associated with these events.
- Look for changes in traffic patterns, for example use of new services or new users of services.
- Use historical traffic patterns to explore the extent of a threat.

3) Denial of Service

During a DoS (or DDoS) attack, a large number of spurious requests (that appear to be legitimate) are sent to victim in a very short period of time. A Smurf attack is an example of this kind of attack. A Smurf attack occurs when an attacker sends an ICMP echo request (also referred to as a "ping") packet to the broadcast address of a subnet containing a large number of host machines. The source address of the packet is altered to that of the intended victim, typically a web server. The hosts in the subnet respond with ICMP echo replies to the victim. This quickly overwhelms the network and the victim, effectively denying service.

To detect and defend against this attack:

- **Identify a potential attack:** raise an event when key segment statistics (eg frame and byte rates), deviate from established baselines.
- **Identify the victim:** use the sFlow data to identify top destinations of ICMP echo response.
- **Identify the subnet sourcing the ICMP echo response:** use the traffic forwarding information from sFlow to identify subnets sourcing ICMP echo responses.
- **Identify port through which ICMP echo responses enter site:** use the source port information from sFlow to identify the (router/switch) ports receiving ICMP echo response packets destined for the victim.
- **Block the attack:** install access control filter on router/switch port that will block ICMP traffic from the source subnet to the victim.

Typically large volumes of traffic are generated during an attack. This places a heavy load on the network infrastructure. At these times, it is especially important for traffic data to be available in a timely manner. An sFlow-based system will continue to operate effectively under heavy network load.

sFlow places no appreciable load on network device processing and does not cache data or buffer packets. So even when a switch or router is heavily loaded with forwarding large volumes of traffic, the sFlow monitoring

system will continue to operate. sFlow samples will continue to be taken and forwarded to a central data collector. In addition, the central data collector is unlikely to become overloaded with traffic data in this situation since the volume of data has been reduced through sampling.

sFlow has been designed to use an inherently unreliable data transport (UDP) to forward samples to the central data collector. Sample packet loss results in a slight reduction of the effective sampling rate. In network overload conditions, although some samples will be dropped, some will arrive at the central data collector and the data collector will still be able to build a representative picture of the network traffic.

4) Compromised host identification

In a distributed DoS attack, it is important to understand which hosts have been compromised. This can be done by querying historical traffic information from accumulated sFlow data to generate a list of the top sources of traffic to the identified victim, during the period of the attack. Top sources will be the compromised machines that should be examined for the illicit software that generated the attack traffic.

In one incident at one of this ISP's web hosting facilities, an abnormal frame rate spike of over 300K pps was noticed. Drilling down through sFlow data it was realized that about 20 hosts were generating all of the traffic and it was targeted at one destination. The source addresses were not spoofed, and were within the hosting center. These hosts had all been compromised. The ISP was then able to work with their clients to upgrade these hosts.

5) Intrusion detection example

Using the blanket audit trail that can be built up from accumulated sFlow data it is possible to detect intrusions. A technique used successfully by this ISP is to look for traffic that is out of place. For example traffic in the wrong place at the wrong time, or headed the wrong way (e.g., spoofing, traffic targeted at infrastructure, etc.). The following example is representative:

The ISP was intrigued by a spike in the traffic on a core switch interface (Figure 6 Core Switch Frame Rate) on 23rd December.

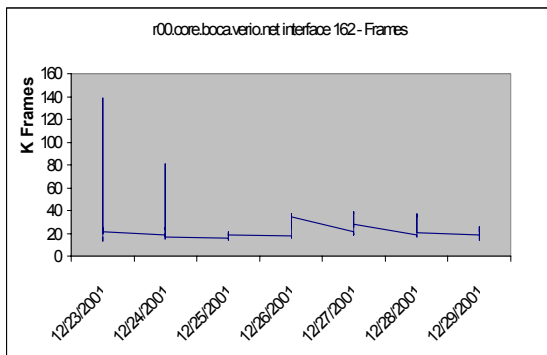


Figure 6 Core Switch Frame Rate

Looking at the top servers (Figure 7 Top Servers) contributing to this traffic, it seemed as if a host, 10.143.81.223, on one of their sites sent about 23GB of data.

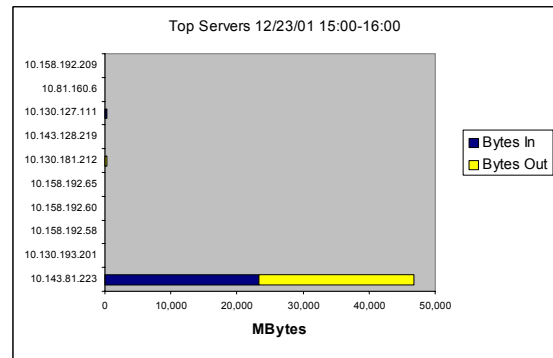


Figure 7 Top Servers

Looking at the top clients of this server (Figure 8 Top Clients), it was determined that all this traffic was being sent to a single destination, 10.34.208.61, with source port = 25 (SMTP).

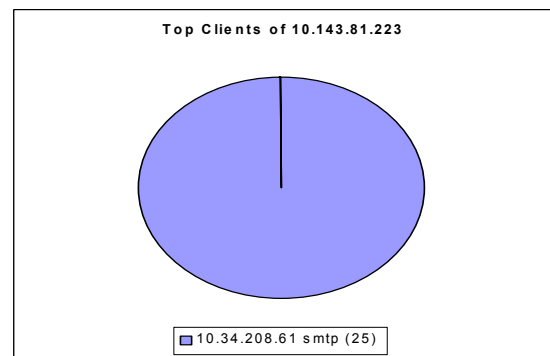


Figure 8 Top Clients

Further analysis (Figure 9 Other Servers Impacted) shows that 10.34.208.61 (on the excite@home network) was sweeping around trying the pop3 port on several other servers.

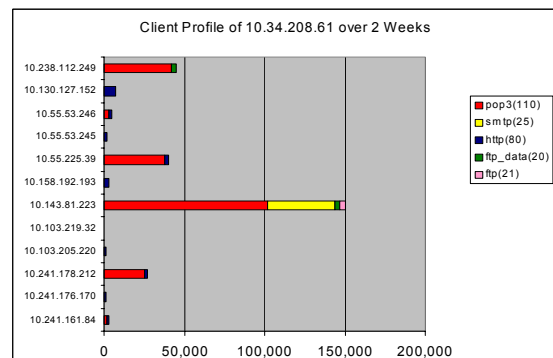


Figure 9 Other Servers Impacted

10.34.208.61 was thought to be a hacker, and that he used port 25 to transfer the entire contents of the disk from his victim (presumably because he thought port 25 would get through firewalls and not attract attention).

V. CONCLUSION

In this paper we have described a technique for continuous, network-wide monitoring of network traffic using packet-based sampling. We have given examples of its large-scale deployment in both an enterprise and an ISP setting. We have shown, through examples, how packet-based sampling for traffic monitoring can be used effectively for defense against both internal and external security threats.

VI. REFERENCES

- [1] Phaal, Panchen, and McKee, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks" RFC 3176, September 2001
- [2] Moore, Savage, and Voelker: "Inferring Internet Denial of Service Activity" Published in proceedings of the 2001 USENIX Security Symposium.