Network Monitoring & Forensics

Jim Irving

Agenda

Network Forensics

- Usefulness
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Host Forensics

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

Introduction

- Network forensics is the capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents.

- Course Goal: To give the student a broad understanding of the main types of network forensic data gathering and an introduction to low level concepts necessary for a proper understanding of the task of performing network forensics. After completion, a student should be able to plan and execute a reasonable network monitoring program and use the gathered forensic data to perform a wide range of investigations.

Benefits

- Why do you care
  - If this isn’t in your toolbelt already, you’ll get a lot of new capabilities when you go on a project.
  - If you’re already seasoned, you can learn from everyone else here.
- Why do I care
  - The Socratic method works.

Disclaimer

The information and views presented during this course concerning software or hardware does not in any way constitute a recommendation or an official opinion. All information presented here is meant to be strictly informative. Do not use the tools or techniques described here unless you are legally authorized to do so.

Agenda

Day 1

- Agenda and motivation
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Day 2

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

Performing Network Forensics What do we need to know?

- What does our network even look like?
- Are we being attacked?
- Is anything compromised?
- How did it get compromised?
- Where are the attacks coming from?

Performing Network Forensics What do we have to work with?

- Loads of recorded network data (PCAP and flow)
- Logs and alerts from security products
- Logs from applications

Main types of forensic data

- We’ll be grouping forensic data into three main data types based on the tools and analysis techniques used
  - Full packet capture (PCAP)
  - Flow data (netflow, IPFIX, etc.)
  - Log / alert data (giant text files)

Forensic Data Type #1
Full Packet Capture (PCAP)

- A **full copy*** of a set of packets travelling over the network
- The most complete form of monitoring possible
- Takes up a lot of space

*it's possible to do partial captures, too

Forensic Data Type #2
Flow Data

- Records of **conversations** on the network
- Stores info such as time, duration, number of packets, total bytes sent, received, etc.
- Does not contain any application layer data
- Good for understanding how data flows on your network quickly

Forensic Data Type #3
Log/Alert data

- Any text that gets written to a file that we can monitor
- Some of it is very important (firewall alerts, availability alerts, etc.) and some of it is less so
- We have to set up things to produce GOOD alerts
- There are a lot of log sources, so some sort of management is preferable

Forensic Bonus Data
People

- This is when someone comes up to you and tells you that they can’t connect to the network, the mail server is down, etc.
- Pretty darned close to real time
- Hard to digitize…

Forensic Data Type Comparison
How do they differ?

	Collection	Storage	What it can reveal	Tools used to Analyze	Typical use
PCAP	Done by machines on the network, taps, and anything that can read 1's and 0's off the network	Consumes lots of disk space. For a project of any size, you'll have to spend money on a storage solution.	**Exactly** what went across the network.	Wireshark, Firewalls, Content Filters, etc.	Deep dive, finding out exactly what commands were issued and how compromises occurred.
Flow	Done by apps on computers on the network or by decent routers	Low space requirements, so it's easy. Generally unified for large networks.	Patterns about conversations, amount of data sent, time, etc.	Silk, Argus, etc.	Retrospective analysis, finding attackers and compromised machines.
Log/Alert	Done by whatever app creates them, wherever it's set to write them.	Generally either left where they were created or consolidated by a log manager or SIEM	Events that occur and are noticed by some piece of software, e.g. attacks, outages, etc.	Splunk, Arcsight, SIEM's	Alerting us to major problems when they occur (or as soon as our log handling methodology shows it to us)

So what do we capture and when?

- Whatever they'll *let* you capture
    - A lot of times the people/systems that you're working with will be totally opposed to you actually *using* the network for anything because the world might end or people might explode. I'll try to give you ways to work your way around this.

So what do we capture and when?

- First get your easy wins
  - Turn on flow data recording on your switches and routers and pump it to some machine.
  - Figure out what log and alert sources are already present and get them into a log manager.

Now you’ve got some flow data and some log/alert data! For free(-ish)!

So what do we capture and when?

- Find out what you’re missing
  - Look at your network diagram and if there’s any part where you’re not getting data from, toss a sensor out there.
- Look at your data and find trouble spots
  - Find events/hosts of interest by analyzing the flow and log data that you’re getting. (More on how to do this later.)

So what do we capture and when?

- Increase monitoring in trouble spots
  - Grab PCAP data from links where you think compromises are occurring.
  - Set up IDS/SIEM/etc. products to produce alerts tailored to the problems you see.
  - Throw host based monitoring apps on suspect machines.

So what do we capture and when? Breakdown

- Log/alert data: Whenever possible, and particularly once you’ve tweaked your alerts.
- Flow data: Whenever possible. It’s easy to capture and easy to work with.
- PCAP data: When you **need** to look closer than flow or log/alert data allows **OR** when you have tons of resources to blow on disk space.

How you'll typically start an investigation

SIEM pops up an alert to your screen, fellow coworker, cell phone, etc saying "Something is horribly wrong on host X!"

You then go look at other logs on host X. Maybe you find something scary. Maybe you can't see the forest for the trees.

Then you open up your flow data for the time in question. See any patterns? Identify suspicious conversations, capture the packets (if you can) and investigate further. Mount some sort of defense against whatever you find.

OR

How you'll typically start an investigation

Somebody hands you a big pile of PCAP or flow data.

Put it through an app to create flow data or IDS alert data (if you don't have it already)

Look for patterns using some analysis tool. Focus down to specific data using those patterns or human reports of problems and get as close to the problem as possible.

Figure out what kind of monitoring you need to get the data you truly need to find the problem, catch the bad guy, or get the conviction. Then go deploy it, assuming you can get client buy-in. (or… create ticket, walk away)

How we’re going to learn this

We’ll be exploring the data types starting at the most finely grained (PCAP) and working up, so that we’ll better understand the limitations of each type, even though in a real investigation, you’d end up using the data in the reverse order.

Agenda

Day 1

- Agenda and motivation
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Day 2

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

PCAP data
Things to think about

PCAP is a straight copy of ALL* network traffic that flows through the pipe for as long as you keep recording. That can be a LOT of data!

- How long do you need to listen?
- Can your NIC capture it fast enough?
- Can your hard drive store it fast enough?
- How long can you listen before you have to free up space?

PCAP data
Line speed and storage

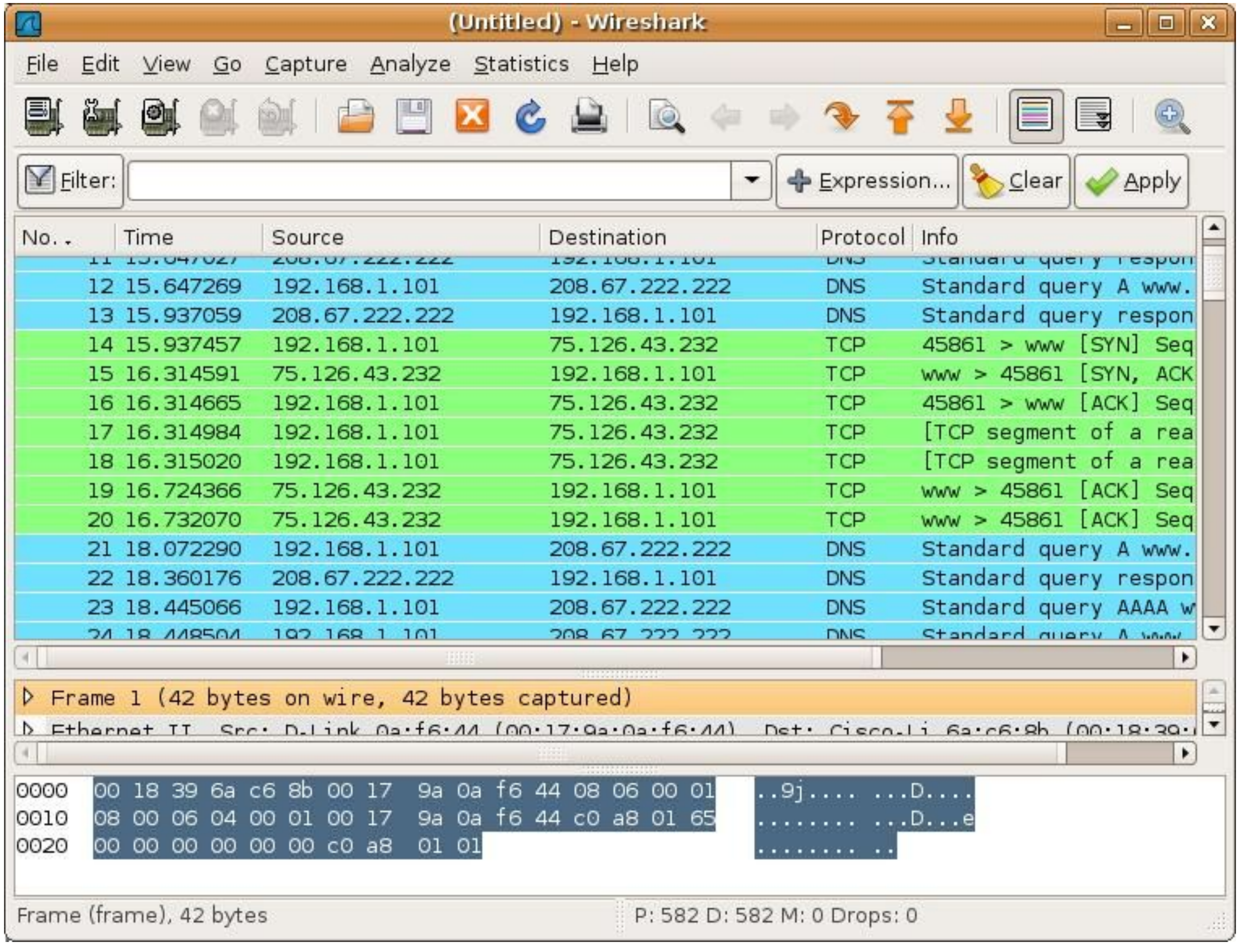
Link type	mb/s	~MB/s	~GB/day
Ethernet	10	1	87
Fast Ethernet	100	10.1	875
OC-12	622.08	63	5,446
Gigabit Ethernet	1,000	101.3	8,755
OC-48	2,488.32	252.1	21,785
10 Gigabit Ethernet	10,000	1,013.3	87,547

Keep in mind, a single width PCI slot can handle, at most, 133 MB/s. Past that you'll need PCI-E NIC's to capture.

Also, commodity hard drives are going to have a maximum write speed around 125 MB/s on a good day.

You'll likely need to either limit your capture time, or spend some money on a RAID solution.

PCAP data
What does it look like?



No.	Time	Source	Destination	Protocol	Info
11	15.647027	208.67.222.222	192.168.1.101	DNS	Standard query respon
12	15.647269	192.168.1.101	208.67.222.222	DNS	Standard query A www.
13	15.937059	208.67.222.222	192.168.1.101	DNS	Standard query respon
14	15.937457	192.168.1.101	75.126.43.232	TCP	45861 > www [SYN] Seq
15	16.314591	75.126.43.232	192.168.1.101	TCP	www > 45861 [SYN, ACK
16	16.314665	192.168.1.101	75.126.43.232	TCP	45861 > www [ACK] Seq
17	16.314984	192.168.1.101	75.126.43.232	TCP	[TCP segment of a rea
18	16.315020	192.168.1.101	75.126.43.232	TCP	[TCP segment of a rea
19	16.724366	75.126.43.232	192.168.1.101	TCP	www > 45861 [ACK] Seq
20	16.732070	75.126.43.232	192.168.1.101	TCP	www > 45861 [ACK] Seq
21	18.072290	192.168.1.101	208.67.222.222	DNS	Standard query A www.
22	18.360176	208.67.222.222	192.168.1.101	DNS	Standard query respon
23	18.445066	192.168.1.101	208.67.222.222	DNS	Standard query AAAA w
24	18.448504	192.168.1.101	208.67.222.222	DNS	Standard query A www.

```
▷ Frame 1 (42 bytes on wire, 42 bytes captured)
▷ Ethernet II, Src: D-Link_0a:f6:44 (00:17:9a:0a:f6:44), Dst: Cisco-Li_6a:c6:8b (00:18:39:
```

```
0000  00 18 39 6a c6 8b 00 17  9a 0a f6 44 08 06 00 01   ..9j.... ...D....
0010  08 00 06 04 00 01 00 17  9a 0a f6 44 c0 a8 01 65   ........ ...D...e
0020  00 00 00 00 00 00 c0 a8  01 01                     ........ ..
```

Frame (frame), 42 bytes   P: 582 D: 582 M: 0 Drops: 0

Source: screenshot of wireshark interface

PCAP data
How we get it

- Network taps
  - Devices that are connected between two other network devices
  - Passively monitors traffic, and reproduces it on one or more monitor ports
  - Available for all media types and speeds

PCAP data
How we get it

- Network taps - keywords
  - Half-duplex: Multiple monitor ports only reproduce one side of the conversation at once
  - Regenerating: Incoming data is copied to multiple monitor ports (for multiple receivers)
  - Aggregating: Receives on multiple ports and combines the data onto a single (full-duplex) monitor port (see problems with oversubscription and timing?)
  - Fail open/closed: when depowered, open lets traffic through, closed does not

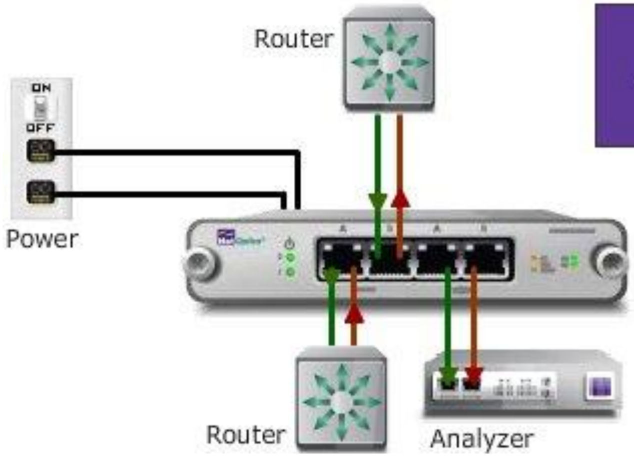
PCAP data
How we get it

- Network taps – dealing with fiber

Fiber taps actually split a portion of the light used to carry the signal, causing the signal downstream to be weaker. When dealing with this, there's a lot more math involved. You will need to calculate a "Loss Budget". This will involve the transmitter power, receiver sensitivity, cable loss, distance, tap characteristics, and anything else that will affect photons. If we end up having lots of extra time, we'll cover this.

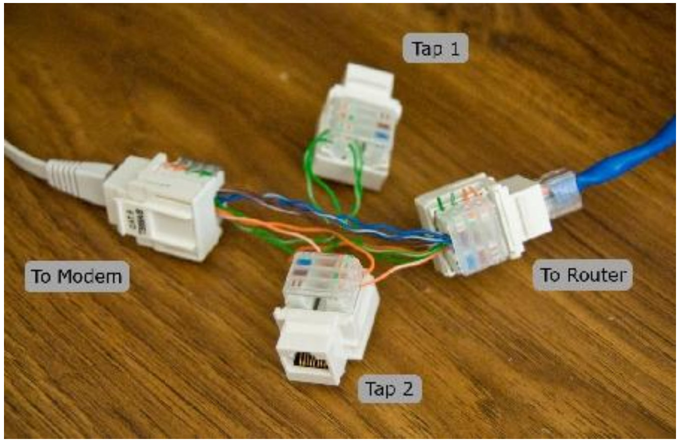
PCAP data
How we get it

- Network taps



Network Copper Tap

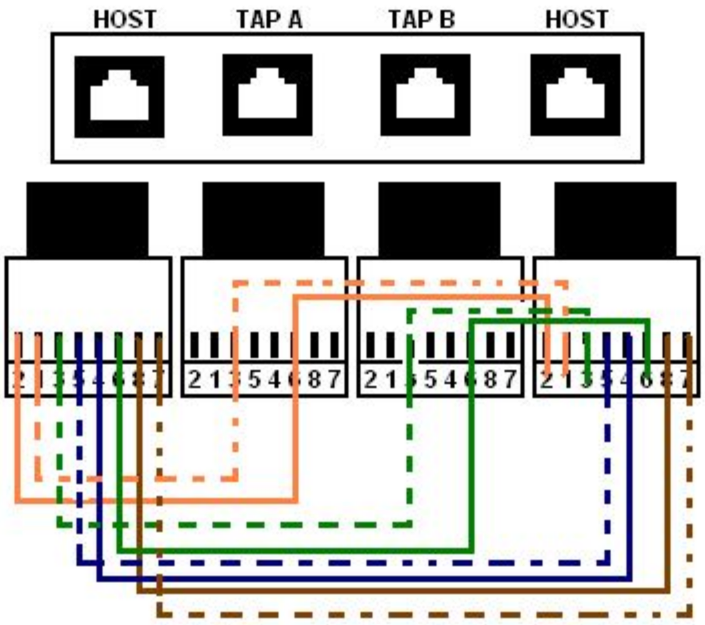
Completely transparent to the network - no point of failure



Source: netoptics.com, hackaday.com

PCAP data
How we get it

- Making a field expedient cat5 tap



Source: thnetos.wordpress.com

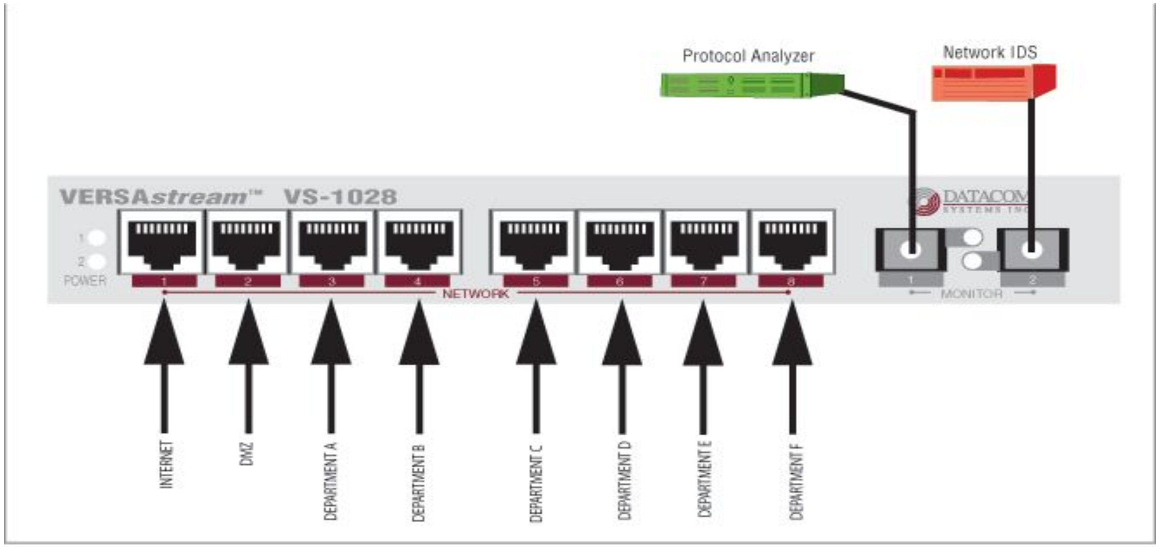
Instructions can be found at http://thnetos.wordpress.com/2008/02/22/create-a-passive-network-tap-for-your-home-network/
Or
http://hackaday.com/2008/09/14/passive-networking-tap/

PCAP data
How we get it

- SPAN ports
  - Ports on most enterprise grade switches/routers which mirror all* traffic on other ports.
  - Will drop packets if there’s not enough bandwidth on the port.
  - You’ll still need a machine connected to it to do the capture.
  - DON’T FORGET TO DO TX *AND* RX!
  - Make your own impromptu SPAN port with the ARP flood trick ☺️

PCAP data
How we get it

- SPAN ports



Source: datacomsystems.com

PCAP data
How we get it

- Direct capture from the NIC on a machine
  - You’ll always do this at some point.
  - Very easy and convenient in low traffic settings. Just start capturing to the hard drive and stop when you feel like it.
  - Storage becomes an issue when
    (traffic * time) > hard drive capacity OR
    (traffic / time) > hard drive write speed
  - Can only see the traffic going to that host (so use taps or SPAN ports to gain visibility)

PCAP data
How we get it

- Direct capture from the NIC on a machine
  - tcpdump
  - wireshark
  - Netwitness
  - etc.

Network coverage – an aside

Network coverage is how much of the traffic on the network that your sensor network can see. You can have different types of monitoring on different parts of the network, but the main idea is to avoid blind spots. This applies to PCAP, flow, logs, and everything else.

Network coverage – an aside

Since different segments of the network carry different traffic, where you decide to place you sensors will determine what you can see.

What would you see on the outside of the border firewall that you wouldn't see inside? What kinds of things do you WANT to see?

Network coverage – an aside

Things to think about

- NAT – solve with placement of sensors
- VPN – solve with placement of sensors or VLAN specific configuration
- Multiple border gateways – solve using channel bonding/aggregation

Network coverage – an aside

On the outside of your firewall, you see the attacks that *didn't* get through in addition to the things that *did*. On the inside of your firewall you see things that actually got through. The outside tells you who's attacking and how. The inside tells you what attacks worked.

Network coverage – an aside

In addition to the amount of the network that’s covered, we can also think about WHEN the network is being covered.

Sometimes you’ll want PCAP data for a couple of hours, but couldn’t handle 24/7. When might that be? Could you perhaps trigger full PCAP for a time based on some event? Absolutely!

PCAP data
Hands on

Now that we know where, why, and how to collect PCAP data, let’s go do some captures.

PCAP data
Doing analysis - Wireshark

Wireshark is your good old fashioned, run of the mill, go-to, protocol analyzing, packet capturing, file carving buddy. Learn to love it.

PCAP data
Doing analysis - Wireshark

- What we’ll be doing today
  - Learning the layout of the interface
  - Capturing PCAP data
  - Looking at the structure of packets
  - Filtering packets to find interesting things
  - Following a TCP session
  - Carving files
  - Reading emails

PCAP data
Doing analysis - Wireshark

- Sources for pcaps
  - http://wiki.wireshark.org/SampleCaptures
  - http://packetlife.net/captures/
  - http://www.pcapr.net
  - http://www.icir.org/enterprise-tracing/download.html
  - Your own machine

PCAP data
Doing analysis - Wireshark

So that’s Wireshark. Pretty nice, huh? When it comes to finding out exactly how your machine got pwned (aka owned, pwnt, etc.), it’s pretty effective.

Also, the functionality of Wireshark can be extended by coding up plugins and decoders, and anything else you want. It’s open source!

PCAP data
Doing analysis - Wireshark

But what if we don’t have time to do all that poking about and sifting through packets? Is there a better way to look through a big pile of PCAP data?

I thought you’d never ask…

PCAP data
Doing analysis - Netwitness

- What we'll be doing today
  - Learning the interface
  - Importing some PCAP data
  - Doing (almost) everything we just did in Wireshark in less time than it took us before
  - Catching things that we might have missed before

PCAP data
Doing analysis - Netwitness

Netwitness is a tool for getting a quick picture of what someone was doing on the network, especially if you’re going after less advanced threats, like insider threats or the average criminal.

Currently there’s a freeware version and a paid version. Give it a try next time you get stuck during an investigation. Often you can catch certain clues via the session based view that you wouldn’t simply by digging through PCAPs.

PCAP data
Doing analysis – Other tools

In addition to sitting down and doing deep dive analysis on PCAP data by hand, we can also run it through automated processes (sometimes even at line speed!) to do all sorts of other stuff. This is how firewalls and IDS work, after all.

Depending on the audience, this is where we discuss our organization’s custom tools ☺️

PCAP data
Generating flow and alert data

- Useful when someone hands you a big wad of PCAP and you have no other data
- Can be done when you’ve got data from before you fielded your flow monitoring or alert generating apps (IDS, firewall, etc.)
- Makes analysis of large data sets easier since it’s faster to look at coarse grained data.
- We’ll cover this when appropriate.

PCAP Data Conclusion

- When you have PCAP you can see pretty much everything.
- It’s very heavy weight whenever you start dealing with enterprise level networks.
- It’s the only way you’ll see what’s being said on the network, but it’s not as good as flow or log/alert data for figuring out what’s important to look at.

Agenda

Day 1

- Agenda and motivation
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Day 2

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

Flow data
Things to keep in mind

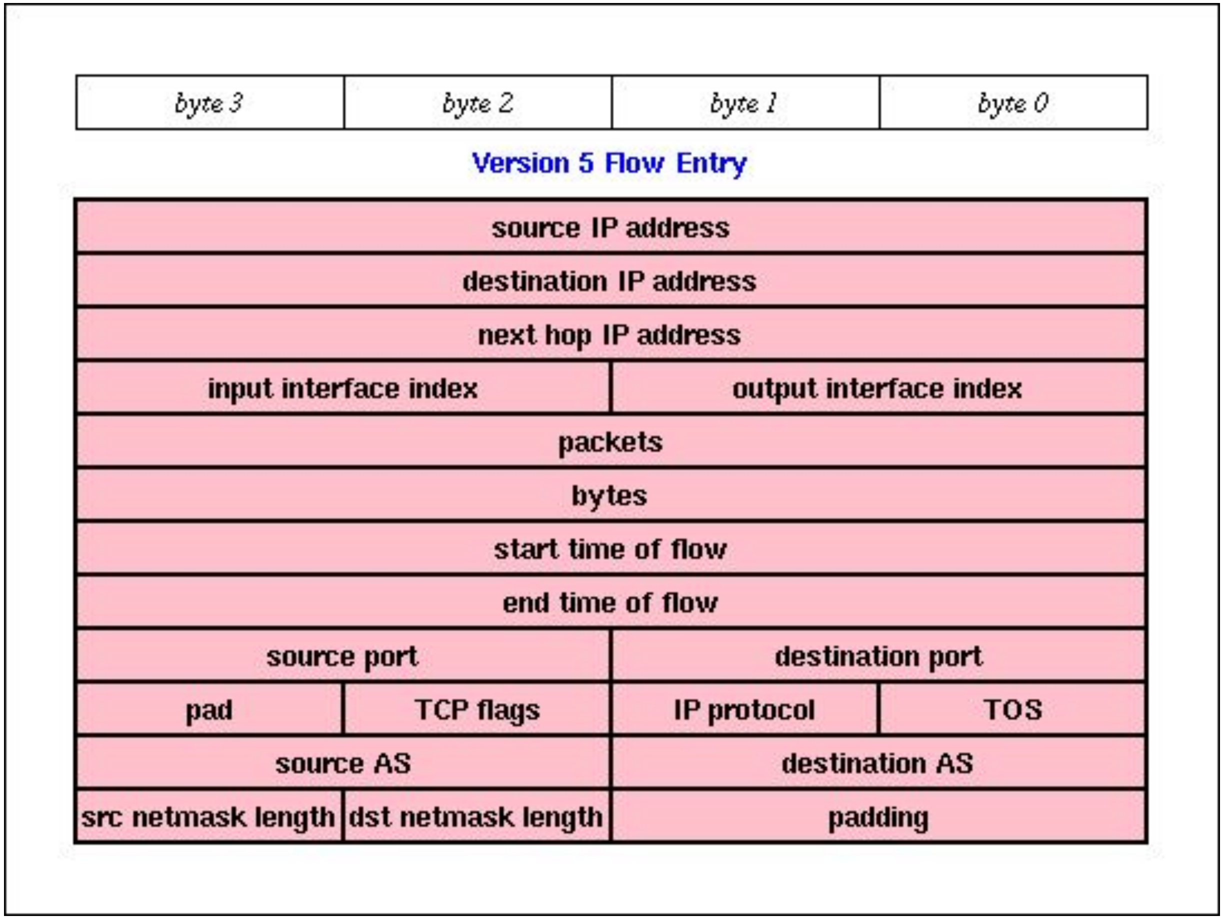
- This is easy data to get, so make sure you do.
- Better used to figure out where to look, than to figure out exactly what happened.
- Even when you’re not on an investigation, you should collect flow data to do baselining.
- Visualization helps a lot.

Flow data
What is flow data?

There’s some variation, but generally a record contains the following:

- Source and dest ip
- Source and dest port
- Protocol
- Start time + (duration | end time)
- # of packets
- # of bytes
- Directionality? Depends on format.

Flow data
Netflow v5 protocol

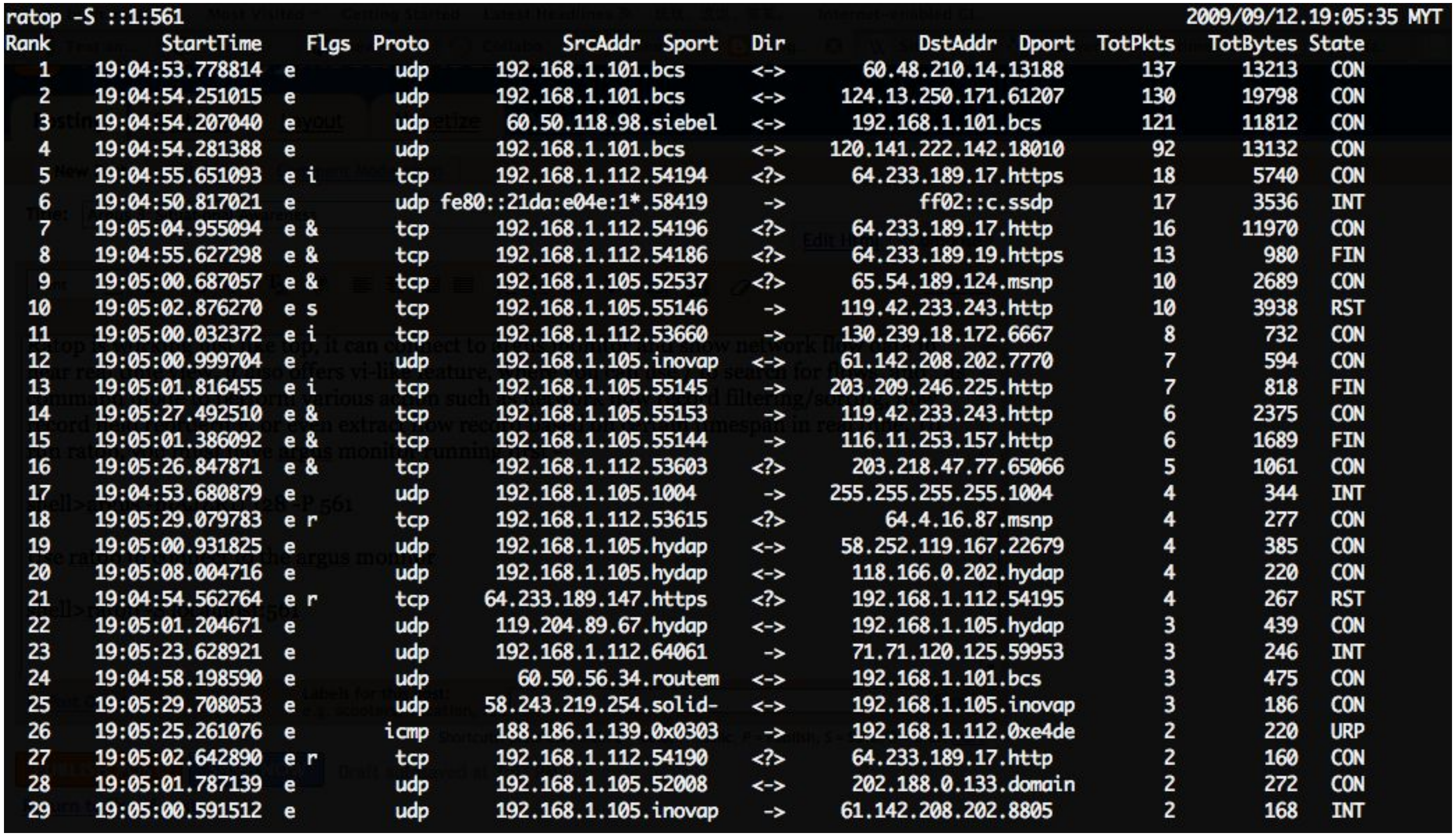


Version 5 Flow Entry

byte 3	byte 2	byte 1	byte 0
source IP address			
destination IP address			
next hop IP address			
input interface index		output interface index	
packets			
bytes			
start time of flow			
end time of flow			
source port		destination port	
pad	TCP flags	IP protocol	TOS
source AS		destination AS	
src netmask length	dst netmask length	padding	

Source: caida.org/tools/utilities/flowscan/arch.xml

Flow data
Command line output



```
ratop -S ::1:561                                                                              2009/09/12.19:05:35 MYT
Rank      StartTime    Flgs  Proto         SrcAddr  Sport   Dir          DstAddr  Dport  TotPkts   TotBytes  State
   1  19:04:53.778814  e      udp    192.168.1.101.bcs       <->     60.48.210.14.13188      137      13213    CON
   2  19:04:54.251015  e      udp    192.168.1.101.bcs       <->   124.13.250.171.61207      130      19798    CON
   3  19:04:54.207040  e      udp     60.50.118.98.siebel    <->      192.168.1.101.bcs      121      11812    CON
   4  19:04:54.281388  e      udp    192.168.1.101.bcs       <->  120.141.222.142.18010       92      13132    CON
   5  19:04:55.651093  e i    tcp    192.168.1.112.54194     <?>     64.233.189.17.https       18       5740    CON
   6  19:04:50.817021  e      udp fe80::21da:e04e:1*.58419    ->            ff02::c.ssdp       17       3536    INT
   7  19:05:04.955094  e &    tcp    192.168.1.112.54196     <?>      64.233.189.17.http       16      11970    CON
   8  19:04:55.627298  e &    tcp    192.168.1.112.54186     <?>     64.233.189.19.https       13        980    FIN
   9  19:05:00.687057  e &    tcp    192.168.1.105.52537     <?>      65.54.189.124.msnp       10       2689    CON
  10  19:05:02.876270  e s    tcp    192.168.1.105.55146      ->      119.42.233.243.http       10       3938    RST
  11  19:05:00.032372  e i    tcp    192.168.1.112.53660      ->      130.239.18.172.6667        8        732    CON
  12  19:05:00.999704  e      udp    192.168.1.105.inovap    <->     61.142.208.202.7770         7        594    CON
  13  19:05:01.816455  e i    tcp    192.168.1.105.55145      ->     203.209.246.225.http        7        818    FIN
  14  19:05:27.492510  e &    tcp    192.168.1.105.55153      ->      119.42.233.243.http        6       2375    CON
  15  19:05:01.386092  e &    tcp    192.168.1.105.55144      ->      116.11.253.157.http        6       1689    FIN
  16  19:05:26.847871  e &    tcp    192.168.1.112.53603     <?>      203.218.47.77.65066        5       1061    CON
  17  19:04:53.680879  e      udp    192.168.1.105.1004       ->     255.255.255.255.1004        4        344    INT
  18  19:05:29.079783  e r    tcp    192.168.1.112.53615     <?>          64.4.16.87.msnp        4        277    CON
  19  19:05:00.931825  e      udp    192.168.1.105.hydap     <->     58.252.119.167.22679        4        385    CON
  20  19:05:08.004716  e      udp    192.168.1.105.hydap     <->      118.166.0.202.hydap        4        220    CON
  21  19:04:54.562764  e r    tcp   64.233.189.147.https     <?>      192.168.1.112.54195        4        267    RST
  22  19:05:01.204671  e      udp   119.204.89.67.hydap      <->      192.168.1.105.hydap        3        439    CON
  23  19:05:23.628921  e      udp    192.168.1.112.64061      ->      71.71.120.125.59953        3        246    INT
  24  19:04:58.198590  e      udp      60.50.56.34.routem    <->        192.168.1.101.bcs        3        475    CON
  25  19:05:29.708053  e      udp   58.243.219.254.solid-    <->     192.168.1.105.inovap        3        186    CON
  26  19:05:25.261076  e     icmp    188.186.1.139.0x0303     ->      192.168.1.112.0xe4de        2        220    URP
  27  19:05:02.642890  e r    tcp    192.168.1.112.54190     <?>      64.233.189.17.http         2        160    CON
  28  19:05:01.787139  e      udp    192.168.1.105.52008     <->     202.188.0.133.domain        2        272    CON
  29  19:05:00.591512  e      udp    192.168.1.105.inovap     ->     61.142.208.202.8805         2        168    INT
```

Flow data Directionality

Some types of flow records are unidirectional (SiLK, rw tools), and others are bidirectional (argus, ratools, original flow data).

Unidirectional flow data has a separate record for both sides of the conversation. This is how Cisco NetFlow v5, v9, and IPFIX records are specified.

Bidirectional flow data combines both sides into one record, usually having extra fields for "# of sender packets", "# of destination bytes", and other things that would get muddled by combining two unidirectional flows.

Flow data Directionality

Depending on what you need, you can convert between bidirectional and unidirectional using whatever tool is appropriate to your data set.

Flow data
Cutoff and Aging

Until conversations end, their flow data sits in the router/switch/etc. memory, taking up space (DOS?). So if we’ve got lots of very long lived flows or flows that didn’t end well (FIN ACK) we need to free up that memory and write the flows.

For long flows, we have a configurable time (say 30 minutes) after which we write a record and start a new one. Figuring out how long the flow actually was will require massaging your data.

For broken flows, another cutoff time (maybe 15 seconds?) will clear them out.

Flow data Sampling

When there’s too much traffic for your switch, NIC, or whatever to handle, **sampling** is used to throttle the workload.

Instead of every packet being recorded in a flow (sample rate = 1 out of 1), we take 1 out of N packets, make flow records, and then scale the appropriate values by N.

We will miss flows due to this ☹ but for very large throughputs it’s necessary. Also, N is not always constant over time.

Flow data Formats

And then there are different formats…

Cisco NetFlow v5 and v9 are very common. V5 will only do IPv4, though.

IPFIX is a lot like v9 plus some interesting fields. Open protocol put out by IETF.

sFlow hardware accelerated, forced sampling, mainly an HP thing.

And there are others, but we'll focus on v5/v9 and IPFIX.

Flow data Formats

There isn’t a current standard for how to store flow data on disk, so different software suites will store it differently to suit their search and compression capabilities. Choose your software suite based on what formats it can *consume*, and be prepared to perform a conversion if you switch.

Flow data Capturing

- Switches and routers
  - Flow data is gathered by the network hardware, and then sent over the network to one or more listeners.
  - To set up collection and forwarding, look up instructions particular to your device and the revision of its OS (typically Cisco IOS).
  - Remember, this is going over the network, so it can be intercepted, falsified, or blocked by attackers, outages, and misconfigurations!

Flow data Capturing

- Machines on the network
  - Creates flow data based on what network traffic that machine can see.
  - Can either generate flow data and forward it to another collector, store it locally, or both.
  - Also possible to collect flow data from other machines or network hardware.
  - Eventually your flow data will have to end up somewhere. You want that somewhere to be handy to your analysts.

Flow data
Analyzing with argus

Argus is another popular tool which is *much* easier to deploy, so we'll be using it to do some sleuthing.

- Become familiar with a few of the tools
- Locate a scanning machine
- Detect beaconing
- Find activities by a compromised machine
- Find routing misconfigurations

Flow data
Capturing with SiLK

- YAF – yet another flowmeter
  - Produces IPFIX data from files or network traffic
  - Can write to disk or push out over network
  - Lightweight, easy to install
  - Works well with SiLK tools

Flow data
Capturing – consolidating in SiLK

- rwflowpack
  - Part of the SiLK toolset
  - Designed to receive input from multiple sensors and build a consolidated repository for analysis
  - Just one of the pieces of a full sensor network.

Flow data
Analyzing with SiLK

- SiLK tools
  - Produced by CERT NetSA
  - Relatively easy to use
  - We’ve already been using them and have done a decent amount of writing on how to use them (check my transfer folder)

Flow data
SiLK tools - conclusion

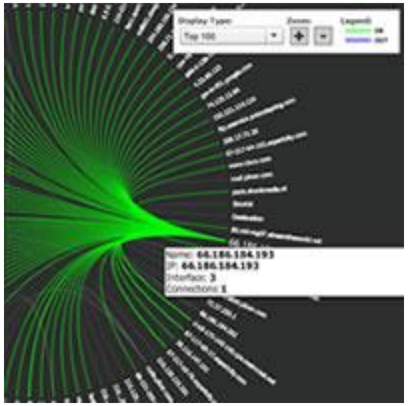
- Free, very powerful, extensible, pretty easy to use.
- Command line tools are great for things that we have running as daemons, but for visualizing flow data we can find a better interface. With the right tools, we can add better visualization.

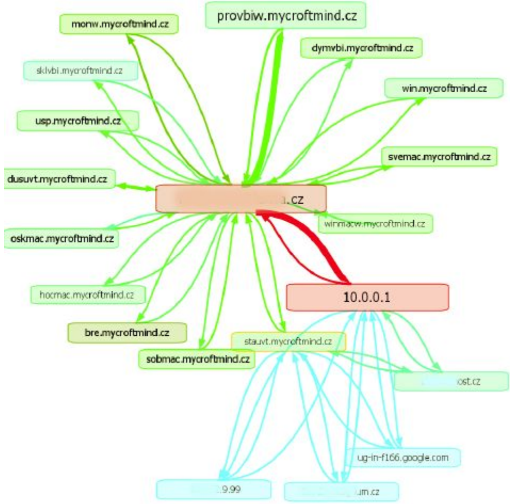
Flow data Visualizing

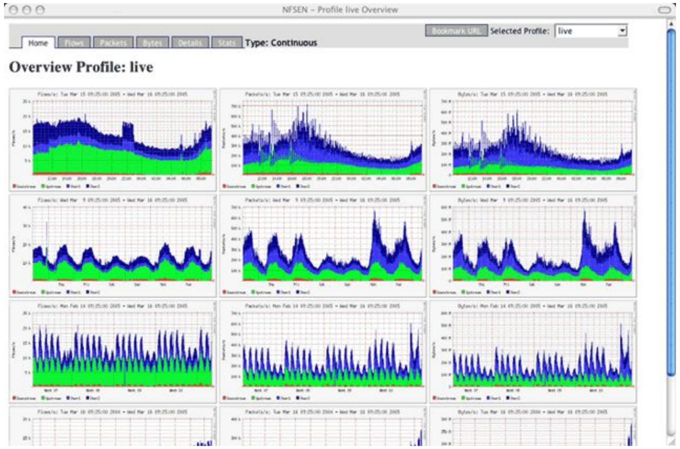
- Open source
  - Afterglow + graphviz: cheap, but too much work to set up
- Free/commercial
  - Scrutinizer: quick and easy, consumes pretty much any flow data, free version is limited to 24 hours of data
  - Lynxeon: belongs in the SIEM category, visualization tool is worth a mention though, 60 day trial

Flow data Visualization

- http://www.networkuptime.com/tools/netflow/
- http://freshmeat.net/search/?q=netflow&section=projects
- TONS more







Source: plixer.com, vizworld.com, networkuptime.com

Flow data
Continuing research

- Flowcon, Centaur Jam, etc.
  - Come join us!
  - Share your tools!
- Statistical anomaly/group detection
  - Complicated math
  - New-ish technology, but worth a look if you've got a pile of netflow data that you're sitting on.

Agenda

Day 1

- Agenda and motivation
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Day 2

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

PCAP reCAP

- Most granular data we can collect
- Takes a lot of resources to gather
- Great for finding out how machines got pwned
- Bad for figuring out what’s going on quickly
- Can be converted into flow and alert data with the right tools

FLOW reFLOW

- Info about conversations on the network
- Cheap and easy to collect
- Quick to analyze with the right tools
- Different analysis suites, formats

Learning styles to use

- More tool use?
- More theory?
- More collaboration!
- You’ve got threats. I’ve got solutions.

Questions about anything up to now?

Agenda

Day 1

- Agenda and motivation
- Intro to forensic data types
- Working with PCAP data
  - What it looks like
  - How to interpret it
  - How to get it
- Working with flow data
  - What it looks like
  - How to interpret it
  - How to get it

Day 2

- PCAP and flow recap
- Working with logs and alerts
  - What they look like
  - How to interpret them
  - Getting them all in one place
  - SIEM’s and their familiars
- Fielding a monitoring solution

Log/Alert data
What are we dealing with?

Logs are any continual text output stored by applications or devices in the process of their functioning.

Alerts are specialized logs produced by something when certain conditions occur that we had the foresight to set an alarm for. If a log is created saying that something we’ve set up a trigger for has happened, then we’ll get an alert.

Log data
Typical sources

- Web server
- Web proxy
- DNS
- Operating system (/var/log/*)
- SMTP
- Whatever you’re using to manage logons
- Building access controls
- HVAC/ICS/SCADA/Power

Alert data
Typical sources

- IDS
- Firewall
- Host based IDS
- SIEM (Security Information & Event Manager)
- Your server uptime and HA (high availability) stuff
- What else?

Typically alerts are being produced because triggers that we've written are being tripped. If you're not getting useful alerts, then you've configured something wrong!

Alert data
Redundant IDS, etc?

- Extra configuration
- Add personnel
- When one dies- “Multiple TippingPoint IPS Malformed Packet Detection Bypass Vulnerability”
- Increased attack surface
- More filtration, more rules, etc.

Alert data
Let’s go set up some triggers

Here’s how you go about getting good alerts

- Find an incident that you want to be alerted about
- Research what went over the network or got written to a log when that incident was occurring
- Write a rule in your IDS or whatever to create an alert when that traffic is seen
- Test your rule
- Continue testing…

Alert data
What will we use as a trigger?

Snort!

- Open source, support packages available
- Basis for Sourcefire appliances
- Very popular, good support among SIMs
- Very robust community providing rules, extensions, add ons, and anything else you can think of
- Rule set subscriptions can be had from Sourcefire, and rules become free 30 days after they're made available to subscribers

Alert data
How Snort works

1. Reads traffic from network
2. Decodes packets
3. Performs stream reassembly
4. Applies filters
5. Upon the first filter match, an alert is generated

Alert data
Writing Snort rules

Fire up your VM’s. Time to go to work.

We’re going to look at how snort rules are written, what alerts look like, and how to write our own rules.

Alert data
Writing better rules

- Write to the vulnerability, not the exploit
- Understand the base rate fallacy
- Inspection chain
- Test and tune your alerts
- Dumbpig, external checking tools, profiling

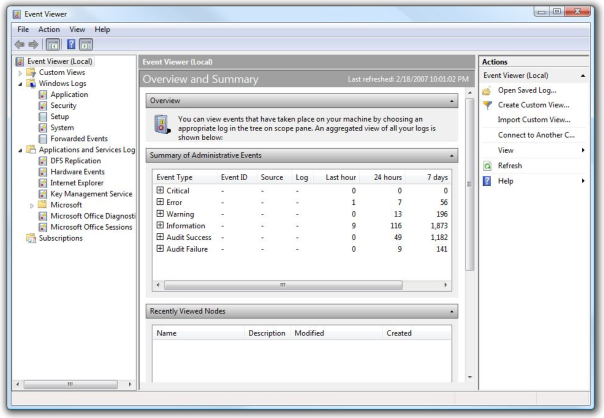
Log/Alert data
Priority of sources

Obviously not all data is equal, so here’s the basic order of which ones you should concentrate on first.

- Alerts from security products (e.g. IDS, SIEM)
- Netflow data, so you can track what those alerts are related to
- OS event logs, so you can see what happened when those alerts were caused
- What else?

Log/Alert data
What does it look like?

- Tons of formats, most of them customizable and flexible, some standards
- Often application specific
- Hard to read straight through, even using search…



Overview and Summary Last refreshed: 2/18/2007 10:01:02 PM

Overview

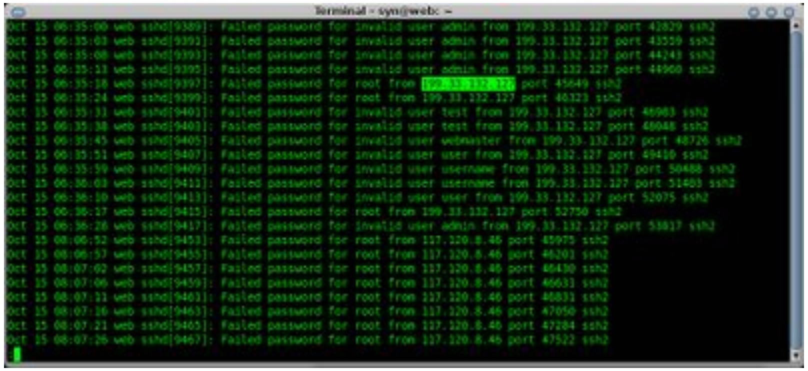
You can view events that have taken place on your machine by choosing an appropriate log in the tree on scope pane. An aggregated view of all your logs is shown below:

Summary of Administrative Events

Event Type	Event ID	Source	Log	Last hour	24 hours	7 days
⊞ Critical	-	-	-	0	0	0
⊞ Error	-	-	-	1	7	56
⊞ Warning	-	-	-	0	13	196
⊞ Information	-	-	-	9	116	1,873
⊞ Audit Success	-	-	-	0	49	1,182
⊞ Audit Failure	-	-	-	0	9	141

Recently Viewed Nodes

Name	Description	Modified	Created



```
Oct 15 06:35:00 web sshd[9389]: Failed password for invalid user admin from 199.33.132.127 port 42829 ssh2
Oct 15 06:35:03 web sshd[9391]: Failed password for invalid user admin from 199.33.132.127 port 43559 ssh2
Oct 15 06:35:08 web sshd[9393]: Failed password for invalid user admin from 199.33.132.127 port 44243 ssh2
Oct 15 06:35:13 web sshd[9395]: Failed password for invalid user admin from 199.33.132.127 port 44900 ssh2
Oct 15 06:35:18 web sshd[9397]: Failed password for root from 199.33.132.127 port 45649 ssh2
Oct 15 06:35:24 web sshd[9399]: Failed password for root from 199.33.132.127 port 46323 ssh2
Oct 15 06:35:31 web sshd[9401]: Failed password for invalid user test from 199.33.132.127 port 46983 ssh2
Oct 15 06:35:38 web sshd[9403]: Failed password for invalid user test from 199.33.132.127 port 48048 ssh2
Oct 15 06:35:45 web sshd[9405]: Failed password for invalid user webmaster from 199.33.132.127 port 48726 ssh2
Oct 15 06:35:51 web sshd[9407]: Failed password for invalid user user from 199.33.132.127 port 49410 ssh2
Oct 15 06:35:59 web sshd[9409]: Failed password for invalid user username from 199.33.132.127 port 50488 ssh2
Oct 15 06:36:03 web sshd[9411]: Failed password for invalid user username from 199.33.132.127 port 51403 ssh2
Oct 15 06:36:10 web sshd[9413]: Failed password for invalid user user from 199.33.132.127 port 52075 ssh2
Oct 15 06:36:17 web sshd[9415]: Failed password for root from 199.33.132.127 port 52750 ssh2
Oct 15 06:36:28 web sshd[9417]: Failed password for invalid user admin from 199.33.132.127 port 53817 ssh2
Oct 15 08:06:52 web sshd[9453]: Failed password for root from 117.120.8.46 port 45975 ssh2
Oct 15 08:06:57 web sshd[9455]: Failed password for root from 117.120.8.46 port 46201 ssh2
Oct 15 08:07:02 web sshd[9457]: Failed password for root from 117.120.8.46 port 46430 ssh2
Oct 15 08:07:06 web sshd[9459]: Failed password for root from 117.120.8.46 port 46631 ssh2
Oct 15 08:07:11 web sshd[9461]: Failed password for root from 117.120.8.46 port 46831 ssh2
Oct 15 08:07:16 web sshd[9463]: Failed password for root from 117.120.8.46 port 47050 ssh2
Oct 15 08:07:21 web sshd[9465]: Failed password for root from 117.120.8.46 port 47284 ssh2
Oct 15 08:07:26 web sshd[9467]: Failed password for root from 117.120.8.46 port 47522 ssh2
```

Source: screenshot from Windows Event Viewer

Alert data
Event formats

- CEE – Common Event Expression
- CVE – Vulnerability
- CCE – Configuration
- CWE – Weakness
- CPE – Platform
- CAPEC – Attack Patterns
- …

Log/Alert data
Dealing with disparate data

There's too much text and not enough *con*text. We need a way to get to the important logs and alerts quickly.

That's why we use log managers and SIEM's. They import the logs into one place, give us some pretty graphs, and (hopefully) make sure that the important entries catch our attention quickly.

Log/Alert data
SIM, SEM, SIEM…

- SIM = Security Information Management
- SEM = Security Event Management
- SIEM = Security Information and Event Management

SIM is for bookkeeping, SEM is for correlating data into events, and SIEM is a combo of the two.

Log/Alert data
SIEMs

- Perform event correlation, reduce false positives
- Help filter logs and alerts to bring us the important data quickly under one monitor
- Typically have a method for reading lots of log types
- This is what you have running on a dedicated monitor in your lab for a technician to keep an eye on and call you when it turns red

Log/Alert data
Some common managers/SIEMs

- Splunk: free version will read 500MB/day of logs, has a decent interface to set up log parsing, technically just a log manager

- ArcSight: popular SIEM suite, has its own log manager, could have a class just on Arcsight alone (and there are). BIG player in government and commercial sector, owing greatly to pushbutton compliance auditing.

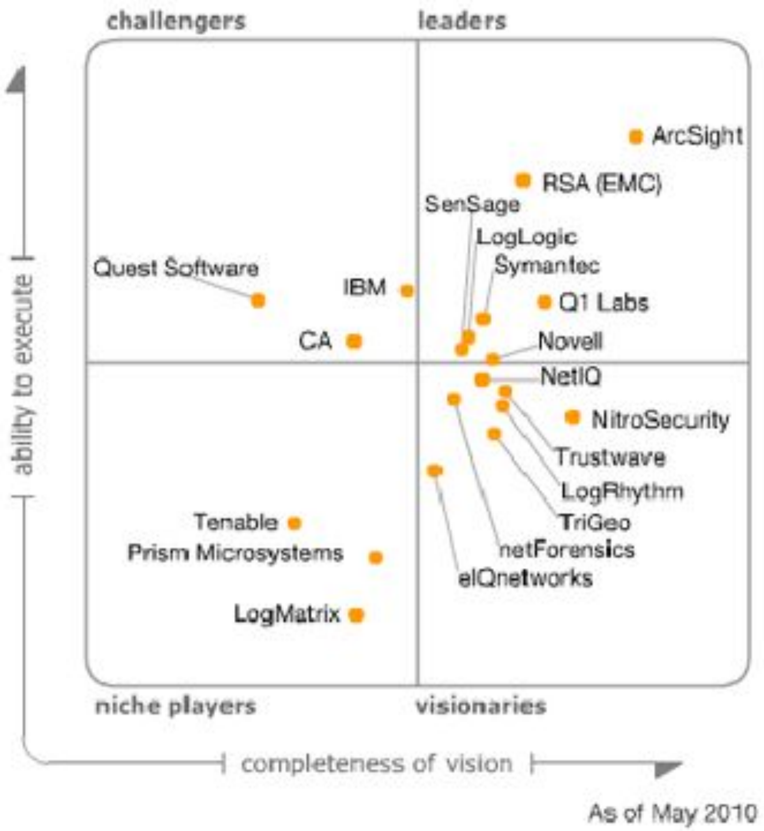
- RSA enVision: another big player, focused on appliances

Disclaimer: the information expressed here is meant only to be informative and does not imply a recommendation

Log/Alert data
Using Splunk

Splunk is common enough that it's worth your time to get to know. So for that reason, we'll now take a quick look through its capabilities and the resources available for learning Splunk 4.0.

Log/Alert data
Some common managers/SIEMs



http://www.gartner.com/technology/media-products/reprints/nitrosecurity/article1/article1.html
Source: Gartner (May 2010)

Log/Alert data
Arcsight event priority

- Recalculated by ESM

- Factors in:

Normalized Severity	S	[0—10]
Model of Confidence & Relevance	MCR	[0—1]
Security History	H	[1—1.3]
Asset Criticality	C	[0.8—1.3]

- Priority = S * MCR * H * C

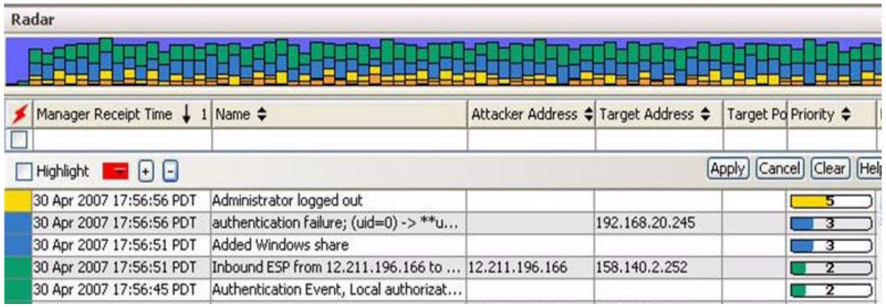
Log/Alert data
Arcsight event priority

- Priority = S * MCR * H * C

- MCR is the only factor that can drop P to 0
  - Fully modeled asset, zero ports, zero vulnerabilities
    - MCR = 0 ☐ Priority = 0

- False positives fed into SIEM force H > 1
  - Avalanche multiplication of false positives

- Worst case: False positives + no asset modeling



	Manager Receipt Time ↓ 1	Name ⬍	Attacker Address ⬍	Target Address ⬍	Target Po	Priority ⬍
☐						
🟨	30 Apr 2007 17:56:56 PDT	Administrator logged out				5
🟦	30 Apr 2007 17:56:56 PDT	authentication failure; (uid=0) -> **u...		192.168.20.245		3
🟦	30 Apr 2007 17:56:51 PDT	Added Windows share				3
🟩	30 Apr 2007 17:56:51 PDT	Inbound ESP from 12.211.196.166 to ...	12.211.196.166	158.140.2.252		2
🟩	30 Apr 2007 17:56:45 PDT	Authentication Event, Local authorizat...				2

Source: arcsight console interface

Log/Alert data
Using SIEMs effectively

- Understand the complexity of the tools you are using and allocate personnel appropriately.
- Standardize what information your organization collects. Prioritize which information you set up collection for.
- Regularly look at your flow data. Don’t depend on the SIEM to see everything.
- Write new alert rules to handle your own particular threats.

Deploying a monitoring solution

What you need to monitor a network will vary greatly depending on the size of the network, its purpose, the threats it will face, the technology used to build it, and countless other things.

Now go to www.ratemynetworkdiagram.com and let's play pin the sensor on the network.

Extended topics (if we have time)

- Privacy/confidentiality laws
- Attacking network monitoring devices
- Evading network monitoring
- Wireless monitoring
- What products have you used and which ones did you like?
- What else?

The End!

- Please give feedback!
- Tell your friends!