An *IT Briefing* produced by SearchSecurity.com

IT Briefing:

Implementing Network Security Monitoring with Open Source Tools

By Richard Bejtlick

SearchSecurity

Sponsored By: Sprint

SearchSecurity

Table of Contents

Implementing Network Security Monitoring with Open Source Tools **Page 3**
By Richard Bejtlick

For inquiries and additional information, contact:
Charley Spektor
Director of Ancillary Products, TechTarget
cspektor@techtarget.com

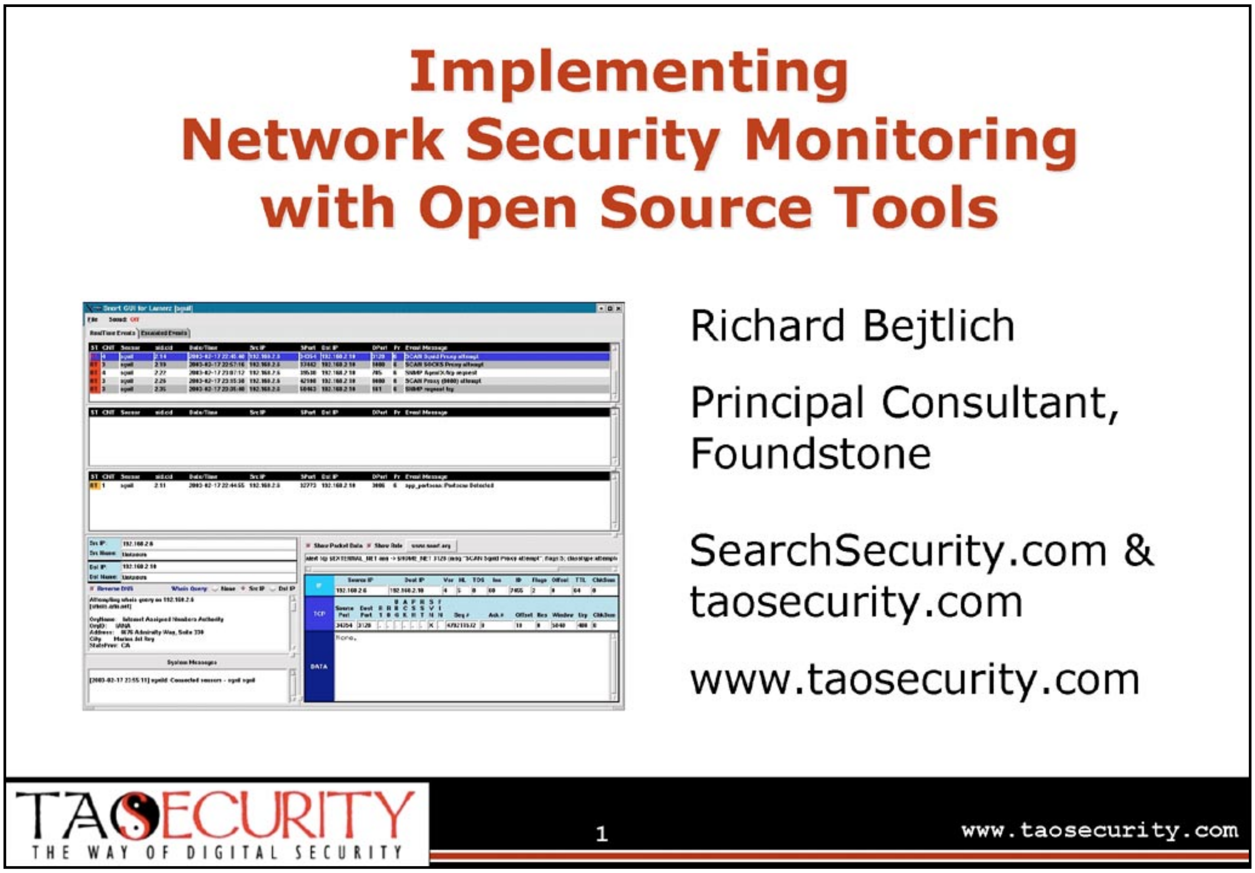
Implementing Network Security Monitoring with Open Source Tools

By Richard Bejtlick

Richard Bejtlick is a principal consultant at Foundstone, where he performs incident response, digital forensics, security training and consulting on network security monitoring. Prior to joining Foundstone in 2002, Richard served as senior engineer for Managed Network Security Operations at Ball Aerospace and Technologies Corporation. From 1998 to 2001, Richard defended global American information assets as a captain in the U.S. Air Force's computer emergency response team. He led the ABSI (phonetic) real-time intrusion detection mission supervising 60 civilian and military analysts. Formally trained as a military intelligence officer, Richard holds degrees from Harvard University and the United States Air Force Academy. He acquired his CISSP certification in 2001. His work appears in Hacking Exposed, 4th Edition, and Incident Response, 2nd Edition, both published by Osborne-McGraw Hill. He is currently writing a book titled the Tao of Network Security Monitoring, which will be finished next year. His homepage is www.TaoSecurity.com.

MODERATOR: Hello and welcome to our SearchSecurity.com webcast, "Implementing Network Security Monitoring with Open Source Tools" with guest speaker Richard Bejtlick. My name is Crystal Ferraro, and I am your moderator.

Are you frustrated by the operation of your intrusion detection system? A solution exists, but it's not found on any vendor's product sheet. The answer is network security monitoring, NSM, a collection, analysis and escalation of indications and warnings that detect and respond to intrusions. NSM is not an intrusion detection system, although it relies on IDS-like products as part of an integrated data collection and analysis suite. NSM is about collecting a full spectrum of data types, events, sessions, full content and statistics needed to identify and validate intrusions.



Implementing
Network Security Monitoring
with Open Source Tools

Richard Bejtlich

Principal Consultant,
Foundstone

SearchSecurity.com &
taosecurity.com

www.taosecurity.com

TAOSECURITY
THE WAY OF DIGITAL SECURITY

1

www.taosecurity.com

Figure 1

Today, guest speaker Richard Bejtlick will briefly explain NSM and introduce several specific tools that can augment your existing detection platforms. While Richard will mention Snort, he will place the free version of Snort in its proper place as an engine to generate intrusion detection event data. Richard will discuss tools to collection session and full content data and will describe how to use the free BSE operating system as a monitoring platform. All these tools can be placed next to your current products and services.

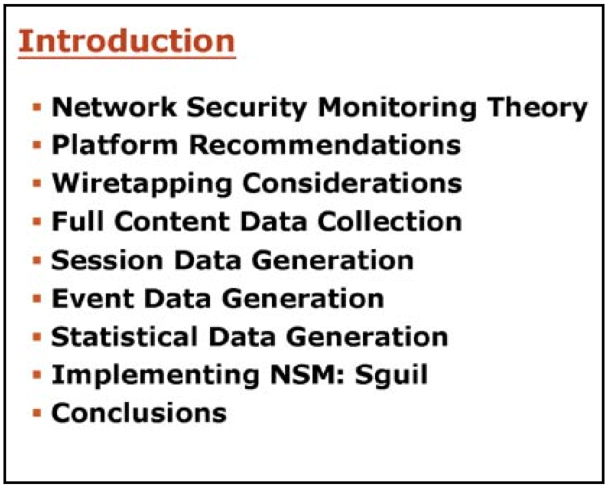
Our guest speaker Richard Bejtlick is a principal consultant at Foundstone, where he performs incident response, digital forensics, security training and consulting on network security monitoring. Prior to joining Foundstone in 2002, Richard served as senior engineer for Managed Network Security Operations at Ball Aerospace and Technologies Corporation. From 1998 to 2001, Richard defended global American information assets as a captain in the U.S. Air Force's computer emergency response team. He led the ABSI (phonetic) real-time intrusion detection mission supervising 60 civilian and military analysts. Formally trained as a military intelligence officer, Richard holds degrees from Harvard University and the United States Air Force Academy. He acquired his CISSP certification in 2001. His work appears in Hacking Exposed, 4th Edition, and Incident Response, 2nd Edition, both published by Osborne-McGraw Hill. He is currently writing a book titled the Tao of Network Security Monitoring, which will be finished next year. His homepage is www.TaoSecurity.com.

Thank you for joining us today, Richard.

BEJTLICK: Thank you very much.

MODERATOR: In case this is your first time participating in a SearchSecurity.com webcast, let me give you an idea of what to expect. To begin, Richard will give a PowerPoint presentation. Slides will be pushed to your screen automatically. If you’d like to make the slides bigger, click the “Enlarge Slides” button. For additional help, click the “Webcast Help” button. After the presentation, I will ask Richard a few questions on today’s topics. You can submit your own questions at any time by clicking on the “Ask a Question” link at the lower left corner of your screen. Richard’s answers will be e-mailed back to you.

Now, we’re ready to turn our attention to Richard Bejtlick and his presentation, "Implementing Network Security Monitoring with Open Source Tools." Take it away, Richard.



Introduction

- Network Security Monitoring Theory
- Platform Recommendations
- Wiretapping Considerations
- Full Content Data Collection
- Session Data Generation
- Event Data Generation
- Statistical Data Generation
- Implementing NSM: Sguil
- Conclusions

Figure 2

BEJTLICK: Thank you, Crystal. I’m happy to be back. I presented on a similar topic in December of last year; I’m happy that SearchSecurity.com decided that I was a worthwhile speaker to bring back. If you joined us for the webcast last year, you know we talked mostly about theory — that is, what was network security monitoring, what were the different types of data that you could collect and where did those ideas come from — things like that.

What I’d like to do today is actually get into the details of how you go about performing network security monitoring and the tools you should use.

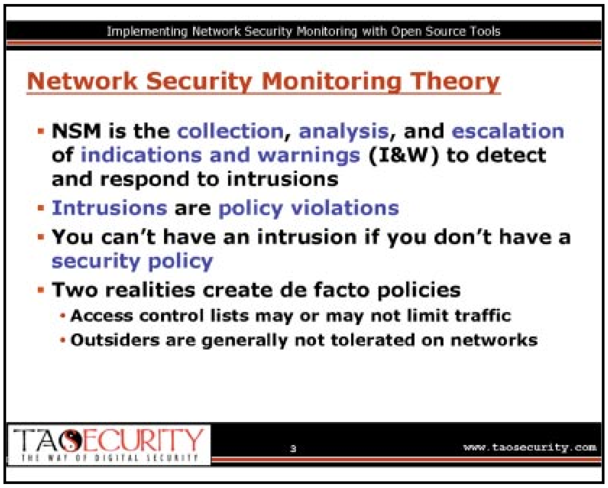
Could I have the next slide, please, Crystal? We should all be looking at a slide that says “Introduction.” Today, I’ll first talk about a little bit of theory, so that if you’ve never heard the term network security monitoring (NSM), you'll have some context for the rest of my presentation. After that, I'll offer some recommendations for platforms — that is, what sort of hardware and operating systems are best suited to hold the tools that you will use to do network security monitoring. I’ll then talk about wiretapping. I call it wiretapping because, well, that’s essentially what it is, in a legal sense and a technical sense. I'll also talk about what sort of legal considerations you might run into and ways you can actually get packets from the wire, hubs, taps, etc. After that, I’ll go into a section where I will discuss the four types of NSM data that can be collected. I'll start with full content, move to session and event data; and conclude with statistical data. For each case, I will give you one or more options for an open source tool. For comparison's sake, I'll also discuss some of the more popular vendor commercial solutions that I've seen or used. After going through the four tools, we'll talk about an open source free products called Sguil that implements one or more NSM theories. I'll conclude by answering some questions.

Next slide, please. So if you didn't participate in the webcast last year I'll provide a quick background on NSM, just to get everyone on the same page. I'd like to define network security monitoring as the collection, analysis and escalation of indications and warnings to detect and respond to intrusions. Now, each one of those words has a meaning. If you have a military background, especially with signals intelligence, some of this probably resonates with you.

NSM is not the same thing as intrusion detection — NSM is more almost about auditing. It's about providing the information that you need in the event of an intrusion to quickly scope and remediate that intrusion. The problem that I see these days with most intrusion detection products is the vendors are very focused on finding events. But once they give you their best guess as to whether or not an event has occurred, they sort of leave you with it. Most customers are more concerned with preventing intrusions. So we've got vendors who are concentrating on finding intrusions, customers who are concerned about preventing intrusions, but neither really has the information you need.

Now, I feel that eventually you're going to have an intrusion. There's just no way you can prevent everything. So once an intrusion occurs and it's detected somehow (by a client, customer, system administrator or an end user who notices that there's something odd going on in the machine), you need enough data to go back, find out what happened and scope that incident as quickly as possible without having to physically touch a thousand different machines. So NSM is about giving you the data that you need to find out what happened. Maybe you won't detect everything. But once you do find out something has happened, how can you quickly go about determining the scope of the intrusion without having to do a whole bunch of host-based forensics?

First, let's define intrusions as policy violations. Immediately you might think, well, most companies or organizations these days are moving towards having some sort of security policy. But if you don't have a security policy, or you don't feel like your security policy is really worth anything, how do you define an intrusion? Well, you can have a policy, either by de facto or de jure. I may be messing that up, but the de facto policy is what you've got in place regardless of whether or not you think you have one. For example, there are two main realities that create de facto policies. One would be having access control, whether from an ACL and a router or some type of ACL on a firewall. If you are doing any type of limiting, you have made some type of decision as to what type of security policy you have.



Implementing Network Security Monitoring with Open Source Tools

Network Security Monitoring Theory

- NSM is the collection, analysis, and escalation of indications and warnings (I&W) to detect and respond to intrusions
- Intrusions are policy violations
- You can't have an intrusion if you don't have a security policy
- Two realities create de facto policies
  - Access control lists may or may not limit traffic
  - Outsiders are generally not tolerated on networks

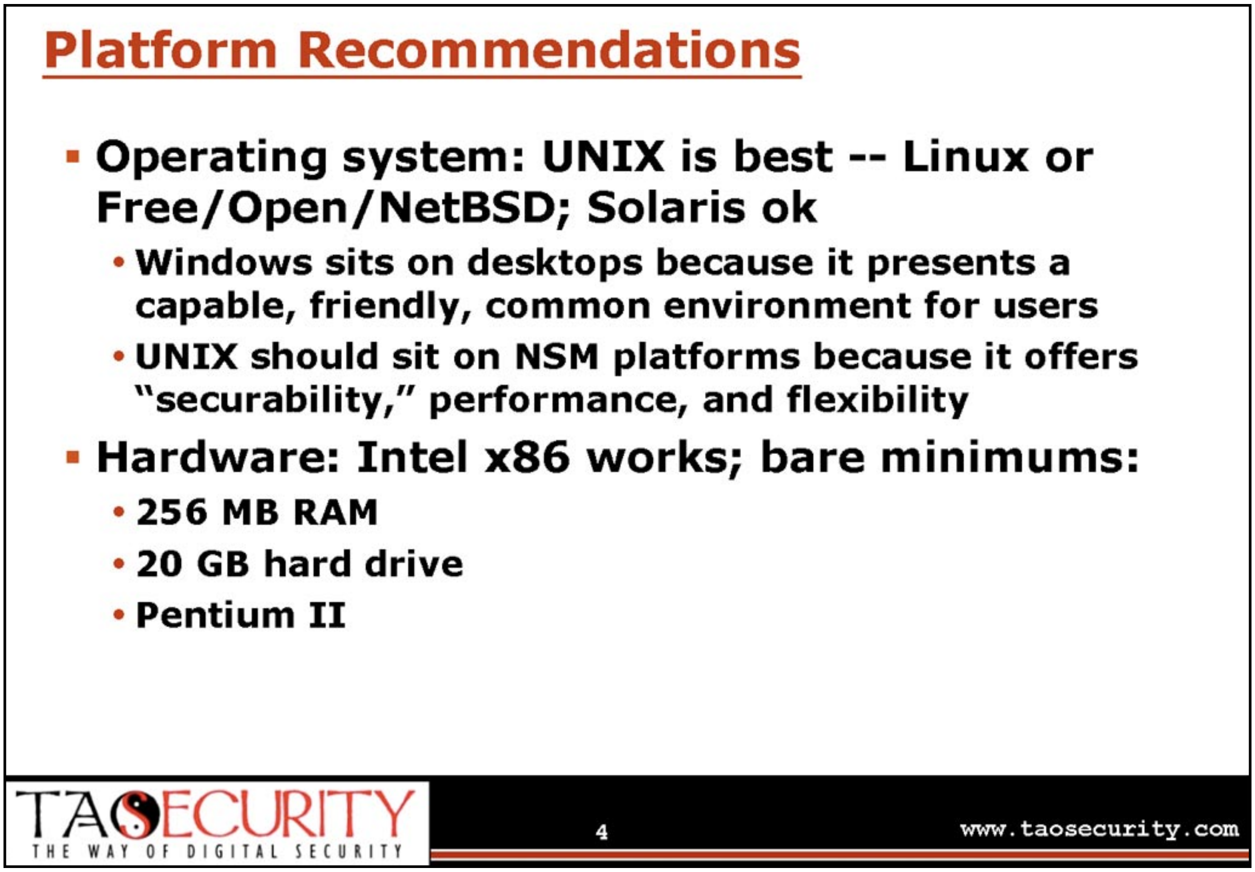
TAOSECURITY THE WAY OF DIGITAL SECURITY 3 www.taosecurity.com

Figure 3

The second reality that most people accept, which creates a de facto security policy, is that most people don't tolerate having intruders on their networks. I have the word outsiders on the slide, because most organizations would not tolerate it if someone was roaming inside their networks. But this isn't always the case. I've had some clients — actually prior to them being clients — who thought it was acceptable to live with an intruder on the network as long as the intruder wasn't too destructive. Most people prefer not to take that route, but, believe it or not, it is an option some people take.

So NSM is not intrusion detection; it's more about auditing. It's also about giving you the information you need to quickly scope an incident and, if possible, discover policy violations.

Could I have the next slide, please? So if you buy into this theory of how we're going to try to scope and remediate intrusions, even in a SIM, you're probably wondering what types of operating systems and hardware are best. I am personally a fan of Unix. In terms of specific versions of Unix, I'm most comfortable with the different BSD operating systems — meaning free, open or netBSD. Linux works as well. If you want to go with a commercial version of Unix, people have good results with Solaris. If you're a Windows person, you may be



Platform Recommendations

- Operating system: UNIX is best -- Linux or Free/Open/NetBSD; Solaris ok
  - Windows sits on desktops because it presents a capable, friendly, common environment for users
  - UNIX should sit on NSM platforms because it offers "securability," performance, and flexibility
- Hardware: Intel x86 works; bare minimums:
  - 256 MB RAM
  - 20 GB hard drive
  - Pentium II

TAOSECURITY THE WAY OF DIGITAL SECURITY 4 www.taosecurity.com

Figure 4

wondering why I don't recommend it. I consider Windows to be the superior desktop operating system; it presents a capable, friendly and common environment for users. But when talking about an OS that can be secured quickly, and offers high performance and flexibility in terms of what you can install or not install, I really don't think you can beat a Unix box.

I've been in emergency situations where I've needed to get a dozen sensors online within hours (typically one or two hours). I've been able to quickly deploy multiple free BSD systems, securely and without having to run any host-based type firewall, simply offering the one or two services that are needed to make that box work. If I had to do that with a Windows box, again without using any type of imaging or ghosting, it would take a long time, and I would have to deploy some type of host-based firewall. I think we found with the recent worms that have been ravaging the Internet that it is almost impossible to have a Windows box defend itself on the Internet without applying some type of third-party or host-based firewall. I mean, you can use group policies and things like that natively to limit Windows exposure, but it's just too difficult to turn off unnecessary services. So I tend to prefer Unix. Plus, in the cases of the BSDs, they typically have very good TCP/IP network stack performance.

As far as hardware goes, more is better in every sense. I have some bare minimums listed. I say that 256 MB of RAM, 20 GB hard drive and a Pentium II will get you pretty far. The thing to keep in mind is, as you start putting your sensors on higher capacity networks and you want to store more days' worth of data, all of those capabilities need to go up.

Obviously, if you're going to try to write every packet to disk on a really busy network, you'll need to store your traffic on 120+ GB disk. Perhaps you'll need a rate array or something to that effect. Again, for a very busy network, you'll need a lot of RAM, because you'll want to keep those packets of memory before they get written to disk. But if you're dealing with a T1 or something like that, I've dealt with networks in boxes that were your garden variety Pentium with a 10 GB hard drive.

Next slide, please. Once you've got your operating system and hardware decided, the next thing is to actually figure out how you get traffic from the wire to that box. There are four main ways you can do that: you can use a hub, a TAP, an inline device, or a SPAN port on a switch.

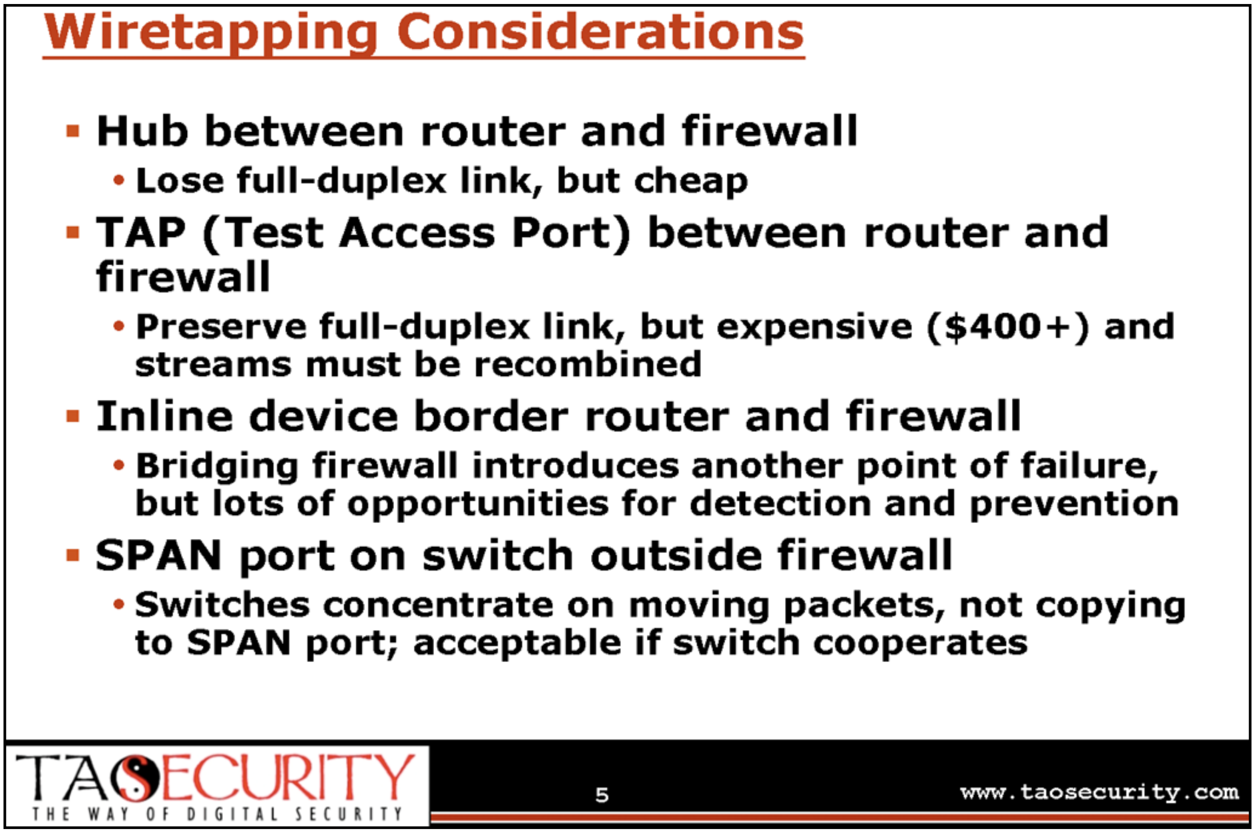
The first option is a hub, which has been popular for many years. You would deploy it as a dumb hub, simply to repeat packets on all interfaces. The problem with that is, if you take the nice full duplex link that is between your router and firewall, and perhaps between two routers or something like that, you'll end up reducing it down to a half duplex link. You'll also introduce a point of failure. So if for some reason the power fails on that hub, or the hub itself has a problem, you've effectively sliced off that part of the network. This has been a problem, believe it or not, with hardware. If you're trying to go this cheap route, you pretty much get what you pay for. I've had hubs in the past, and believe it or not, if you take them off the horizontal by a little bit, they stop passing packets —absolutely crazy. I've also had problems with specific vendors, I won't name which ones I've had problems with; but I will offers my own personal recommendations.

The next option is a TAP. TAP is an acronym standing for test access port. This is a device you can place between the same devices you have used before, the router and firewall, two routers, etc. The nice thing about TAP is that it preserves the full duplex link between those devices. On the downside, though, it is very expensive. It has signal regeneration mechanisms inside, so it costs on the order of $400 or more.

With TAP, there's also the issue of the two streams that come out. There are two actual interfaces, and when you take those interfaces, you need to figure out how to recombine them on the sensor. There are commercial products that will do this and some TAPs that will provide you with a single stream. But let's say you've got a 100 megabit link and you put a TAP in there — when you split the streams, you'll have two 100-megabit lines coming out. There's one vendor I know of who will sell you a TAP with one output coming out. That means that if you ever exceed 50 megabits in either direction, you'll have also exceeded the 100 megabit single line coming out. So I always recommend going for a TAP that has two streams coming out.

In terms of combining those streams, there are two projects that will do this for you. First, there is ether channel bonding from Linux. Second, there is a net graph implementation for FreeBSD, which I've got that outlined on my Web site.

Besides hub and TAP, a third option we have for getting packets off the wire is using an inline



Wiretapping Considerations

- Hub between router and firewall
  - Lose full-duplex link, but cheap
- TAP (Test Access Port) between router and firewall
  - Preserve full-duplex link, but expensive ($400+) and streams must be recombined
- Inline device border router and firewall
  - Bridging firewall introduces another point of failure, but lots of opportunities for detection and prevention
- SPAN port on switch outside firewall
  - Switches concentrate on moving packets, not copying to SPAN port; acceptable if switch cooperates

TAOSECURITY THE WAY OF DIGITAL SECURITY 5 www.taosecurity.com

Figure 5

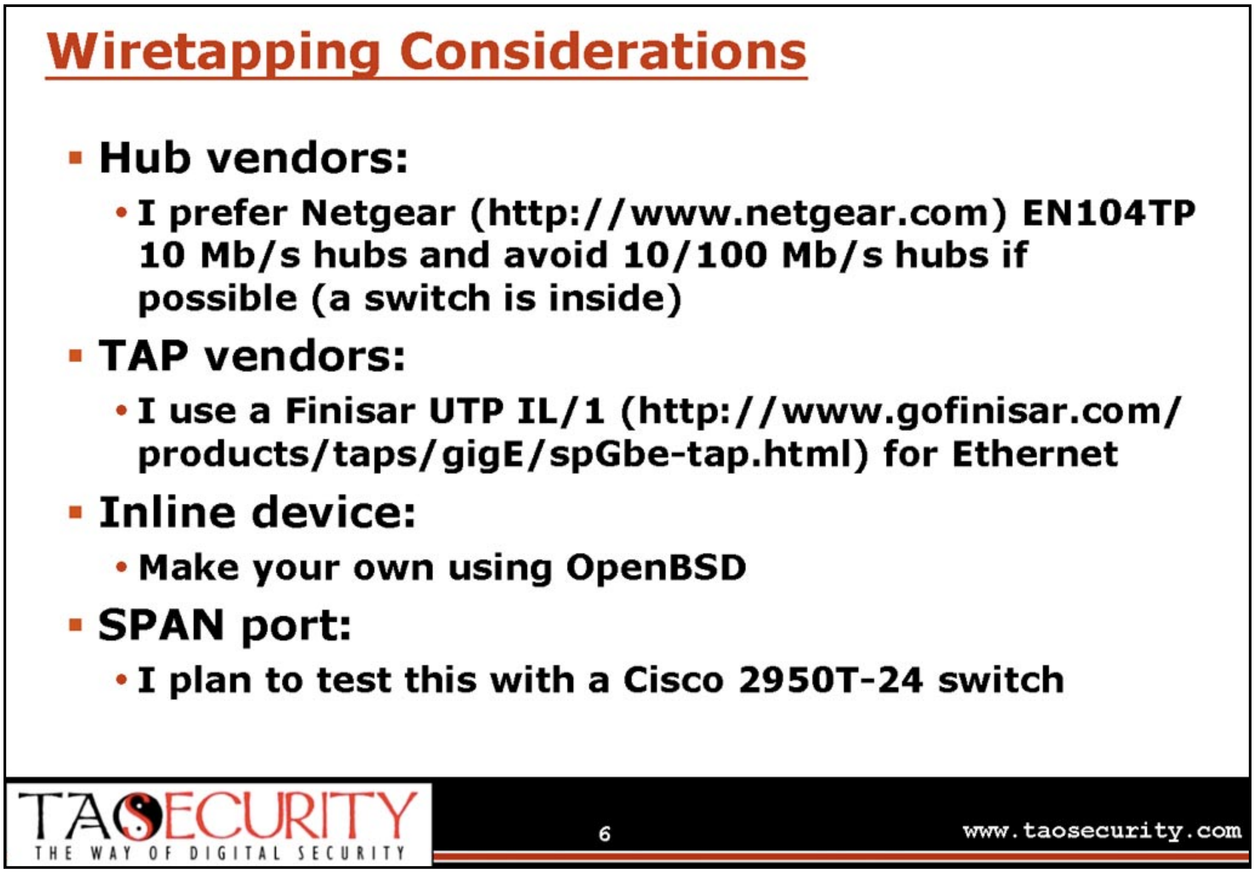
device. This is an opportunity to do not only do some monitoring, but also get additional access control. I'll get briefly into the whole IDS, IPS field in the questions, but with this inline device, you not only can snip traffic, but you can perhaps control it. Again, this introduces another point of failure, so in some cases it may be your best bet.

Finally, we have the SPAN port, which is typically a port on a switch. For example, I've got a Cisco 2950 switch sitting right next to it. I've designated one port to copy all the traffic that is seen on other ports I've specified. If you've got one of these switches sitting around and it's in a good location, this can be a good solution. Keep in mind though that the priority of the first switch is not to copy packets, it's to move packets. So although I haven't personally dealt with this and haven't heard of this happening too often, it is possible that the switch could drop packets if there's too much traffic to copy.

Could I have the next slide please, Crystal? So for those four options, I've got some recommendations here. I'm not trying to push any one vendor's products, and I don't have any ties to these vendors, either professionally with Foundstone or on my own. These are just products that have worked for me in the past. On the hub side, I like Netgear. If you can limit yourself to a hub that's a single speed, do that. For example, let's say you're monitoring a T1 at 1.544 megabits per second. You wouldn't need a 100 megabit hub if you're going to be dropping one in line — you could use a 10 megabit hub.

A 10/100 megabit hub is really a switch. There's got to be something in there that handles the two different traffic speeds. So I prefer to use a straight 10 megabit hub. That way, if you know you plugged something into it, it's going to negotiate to 10 megabits if you didn't set the device to come up at 10, so everything is running at the same speed and you won't lose any traffic.

The second device is a TAP. I have here at home a Finisar UTP IL/1. I've given the full link to the product on the slide because they're kind of hard to find on the Finisar Web site. If you're wondering what Finisar is, it's the company that bought Shpmiti. So Shomiti TAP is now Finisar. There are a couple of other companies that make them, but this is the one I've used and it seems to work pretty well. For inline devices, most people tend to use



Wiretapping Considerations

- Hub vendors:
  - I prefer Netgear (http://www.netgear.com) EN104TP 10 Mb/s hubs and avoid 10/100 Mb/s hubs if possible (a switch is inside)
- TAP vendors:
  - I use a Finisar UTP IL/1 (http://www.gofinisar.com/products/taps/gigE/spGbe-tap.html) for Ethernet
- Inline device:
  - Make your own using OpenBSD
- SPAN port:
  - I plan to test this with a Cisco 2950T-24 switch

TAOSECURITY THE WAY OF DIGITAL SECURITY

6

www.taosecurity.com

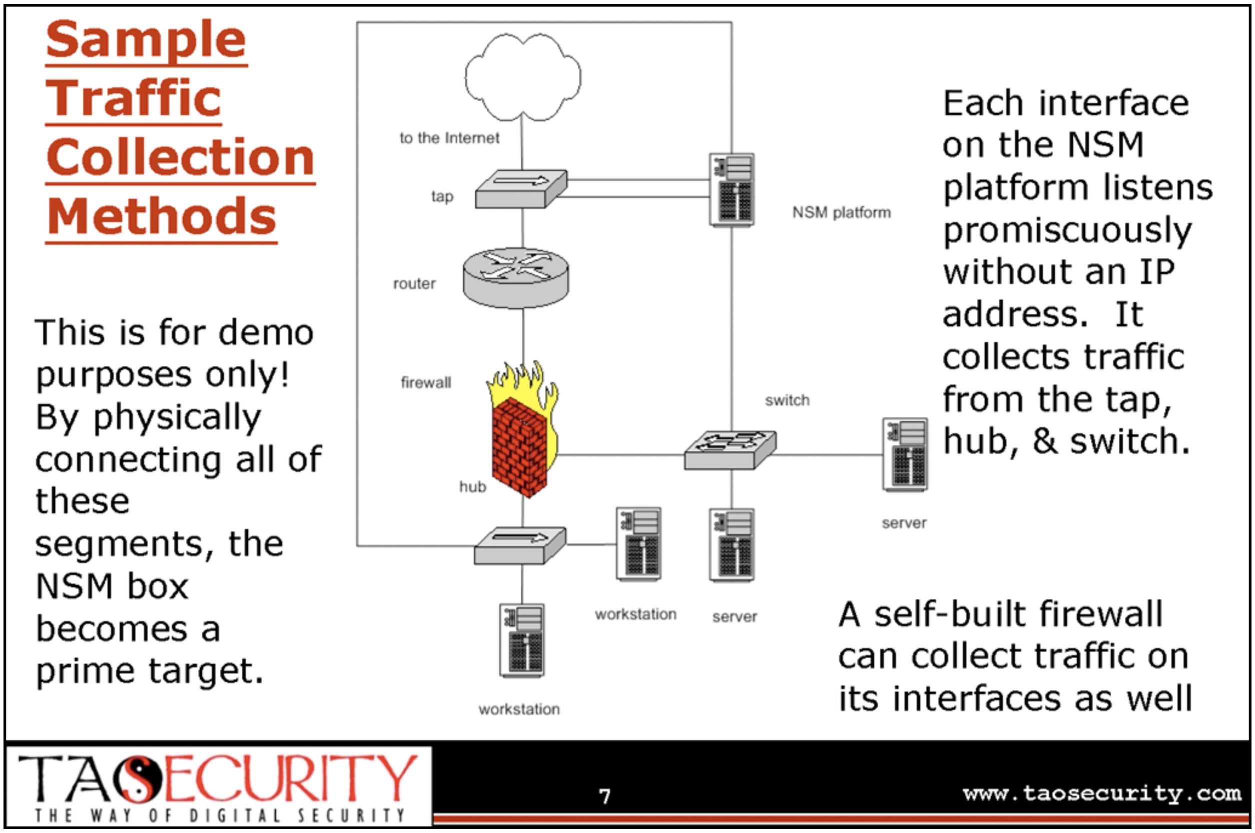
Figure 6

OpenBSD. OpenBSD has very robust, built-in firewalling, packet filtering and packet scrubbing features; so that's a good option. If you're not familiar with OpenBSD and you'd like to learn more about it, there's an excellent book by a gentleman named Michael Lucas called Absolute OpenBSD that I'm reading right now; I hope to have my review up on Amazon shortly. But it is a great so far. It walks you through the installation, tells you what to do; it's really explains how it works. Again, it's called Absolute OpenBSD by Michael Lucas.

Finally, for SPAN port, if you're using any of the commercial-grade Cisco switches (basically if it's a 19 inch 1U type box), chances are it has a SPAN port. There are limitations on what you can do with some of the SPAN ports in terms of directions of traffic and things like that. But, for example, the Cisco 2950T I have is a good choice for that.

Could I have the next slide, please? The next slide is more of a conceptual diagram than an actual deployment diagram; it shows some of these technologies in action. In the upper right-hand corner, we have our NSM platform, which is collecting traffic from three separate spots. There's one line that comes out the top and loops around to the bottom. That shows how you can connect the NSM platform to a hub. You have a couple of workstations that are providing traffic. The port goes out through the firewall. You also have the NSM platform with two lines going off to the left into a TAP. I apologize if some of these diagrams aren't exactly correct, like having what the Cisco versions of what the diagrams should say, but you get the idea.

So you have the TAP between a router and going out to the Internet. At the bottom of the NSM platform, we have lines going into a switch; that's just to depict that the switch could have a SPAN port [audio] traffic to the NSM platform. The thing to keep in mind with all this is that all the interfaces on this NSM platform should listen promiscuously without an IP address. In other words, there should be no way for a person on any of the segments to directly communicate with that NSM platform. If you wanted to communicate with it remotely, I would recommend you add another interface and connect an administrative network to that NSM platform. If you have the luxury to simply walk up to it and log in, that's the best solution. Another option would be to have some type of administrative server behind the firewall with a



Sample Traffic Collection Methods

This is for demo purposes only! By physically connecting all of these segments, the NSM box becomes a prime target.

Each interface on the NSM platform listens promiscuously without an IP address. It collects traffic from the tap, hub, & switch.

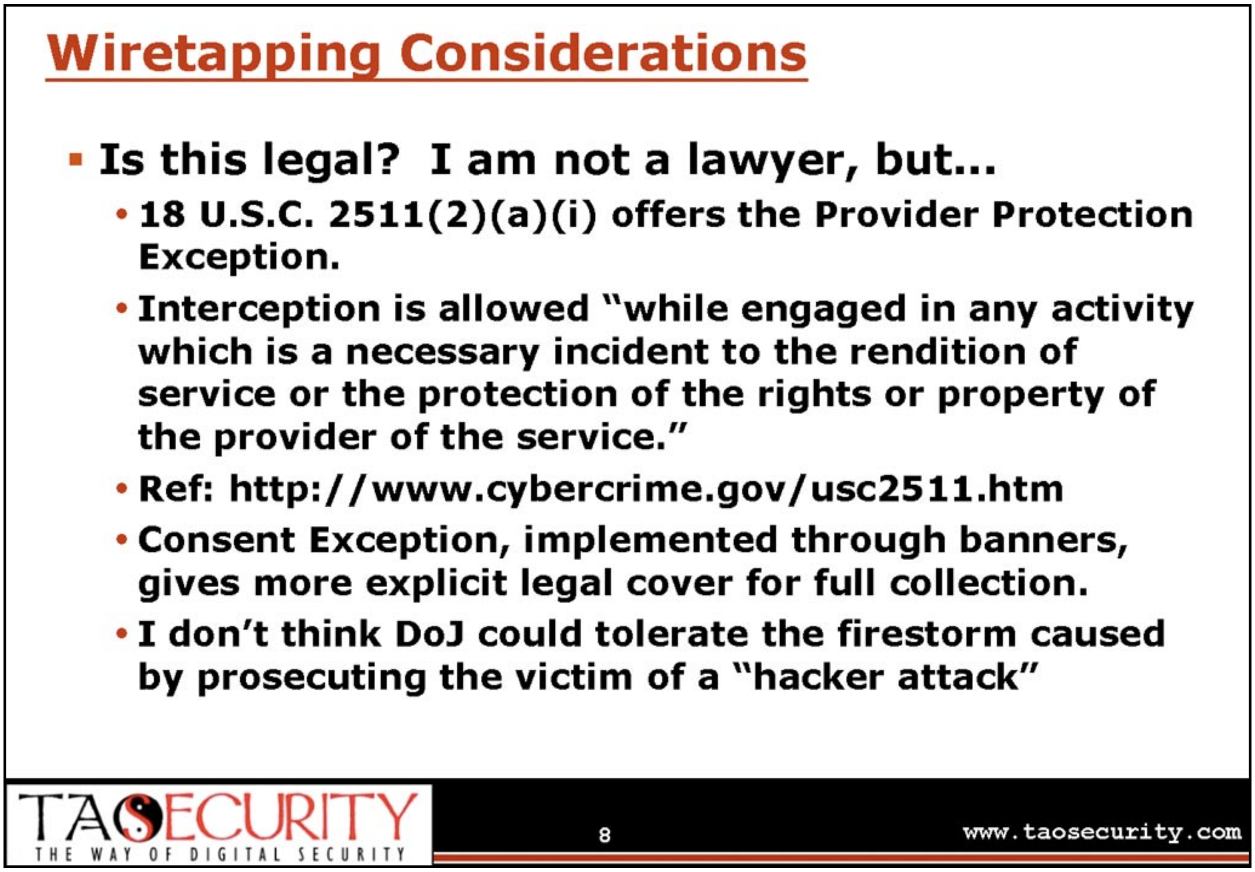
A self-built firewall can collect traffic on its interfaces as well

TAOSECURITY
THE WAY OF DIGITAL SECURITY

7

www.taosecurity.com

Figure 7



Wiretapping Considerations

- **Is this legal? I am not a lawyer, but…**
  - **18 U.S.C. 2511(2)(a)(i) offers the Provider Protection Exception.**
  - **Interception is allowed "while engaged in any activity which is a necessary incident to the rendition of service or the protection of the rights or property of the provider of the service."**
  - **Ref: http://www.cybercrime.gov/usc2511.htm**
  - **Consent Exception, implemented through banners, gives more explicit legal cover for full collection.**
  - **I don't think DoJ could tolerate the firestorm caused by prosecuting the victim of a "hacker attack"**

TAOSECURITY
THE WAY OF DIGITAL SECURITY

8

www.taosecurity.com

Figure 8

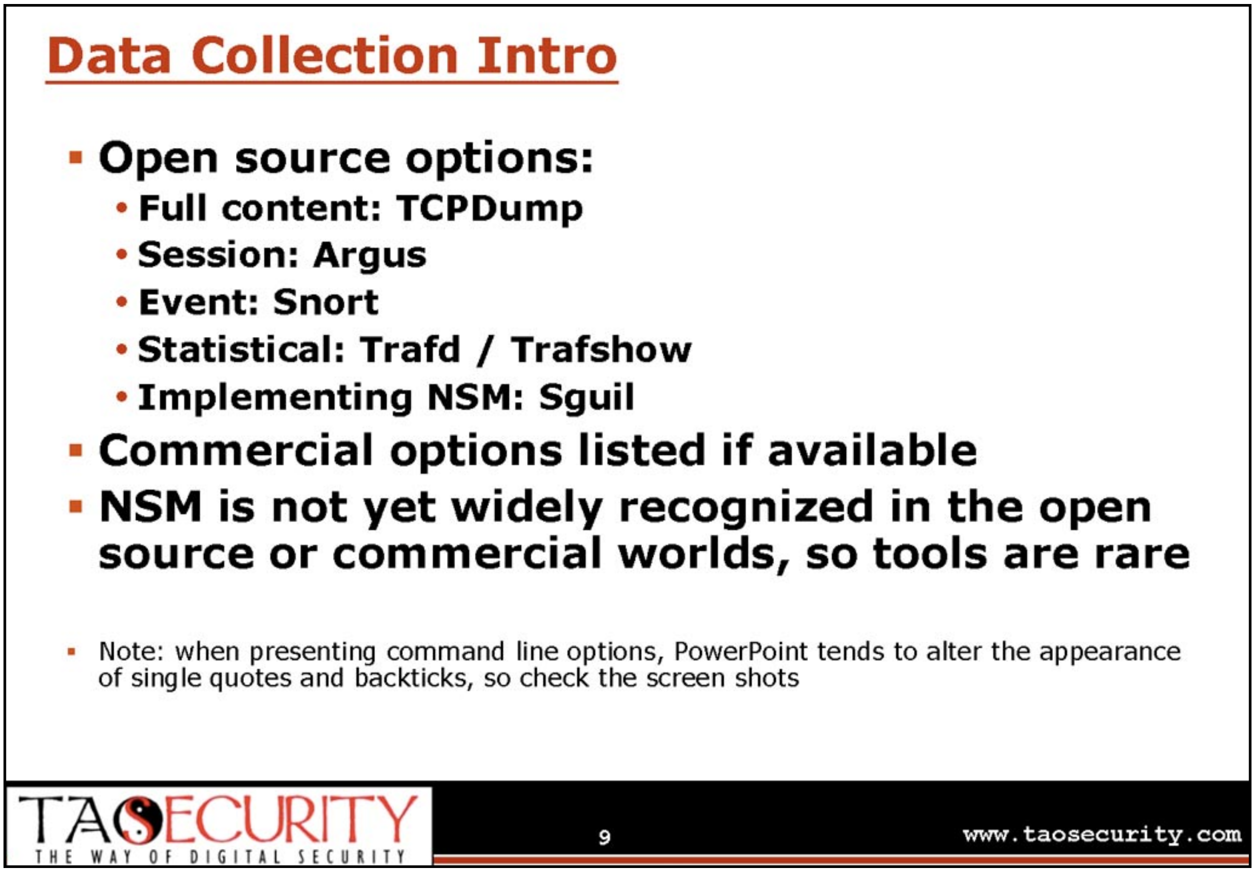
serial cable going into the NSM platform. That limits you in some cases to just looking at a terminal, but it is a more secure option.

Again, I say this is more of a conceptual deployment rather than an actual deployment. If you were to think of some way to potentially compromise that NSM platform once you're on that box, you have visibility to every network that is on this tiny organization that I have created here.

One more thought: if you've created the firewall — actually, if you've created any other devices here — if you built a software-based router, say a Linux based or FreeBSD based router or OpenBSD based firewall, or even gone so far as to create your own open source switch, you can run NSM type tools on any of those devices and collect traffic as well.

Could I have the next slide, please? Now, before we go into tools that collect traffic, I need to mention some legal issues. I have this standard "I am not a lawyer" definition here, but something to keep in mind is that when you do collect traffic in this manner, especially when collecting headers and full content, you are doing a wiretap. I don't know if I'd call them lowest members of the Department of Justice, but there are people that are constantly warning system administrators and network administrators to be careful about the data they collect. A lot of what we have to worry about, at least at the federal level, is 18 U.S.C. 2511 (2)(a)(i); this aspect of the wiretap act provides us with two exceptions under which we can hide and use this cover for not going to jail when we're collecting traffic.

The first one is the provider protection exception. What this says is, if you are engaged in any activity that is necessary to protect the rights and property of that activity, then you're covered. The best way to implement something like this is to have a written security policy that says: In order to protect my company, my university or my organization, I need to collect data of this type — full content data, session data, event data and statistical data. That way, if you're ever busted by the feds for collecting traffic, you can say, look, this is my security policy. I need to do this to protect my organization and I claim protection under the Provider Protection Exception.



Data Collection Intro

- Open source options:
  - Full content: TCPDump
  - Session: Argus
  - Event: Snort
  - Statistical: Trafd / Trafshow
  - Implementing NSM: Sguil
- Commercial options listed if available
- NSM is not yet widely recognized in the open source or commercial worlds, so tools are rare

- Note: when presenting command line options, PowerPoint tends to alter the appearance of single quotes and backticks, so check the screen shots

TAOSECURITY THE WAY OF DIGITAL SECURITY 9 www.taosecurity.com

Figure 9

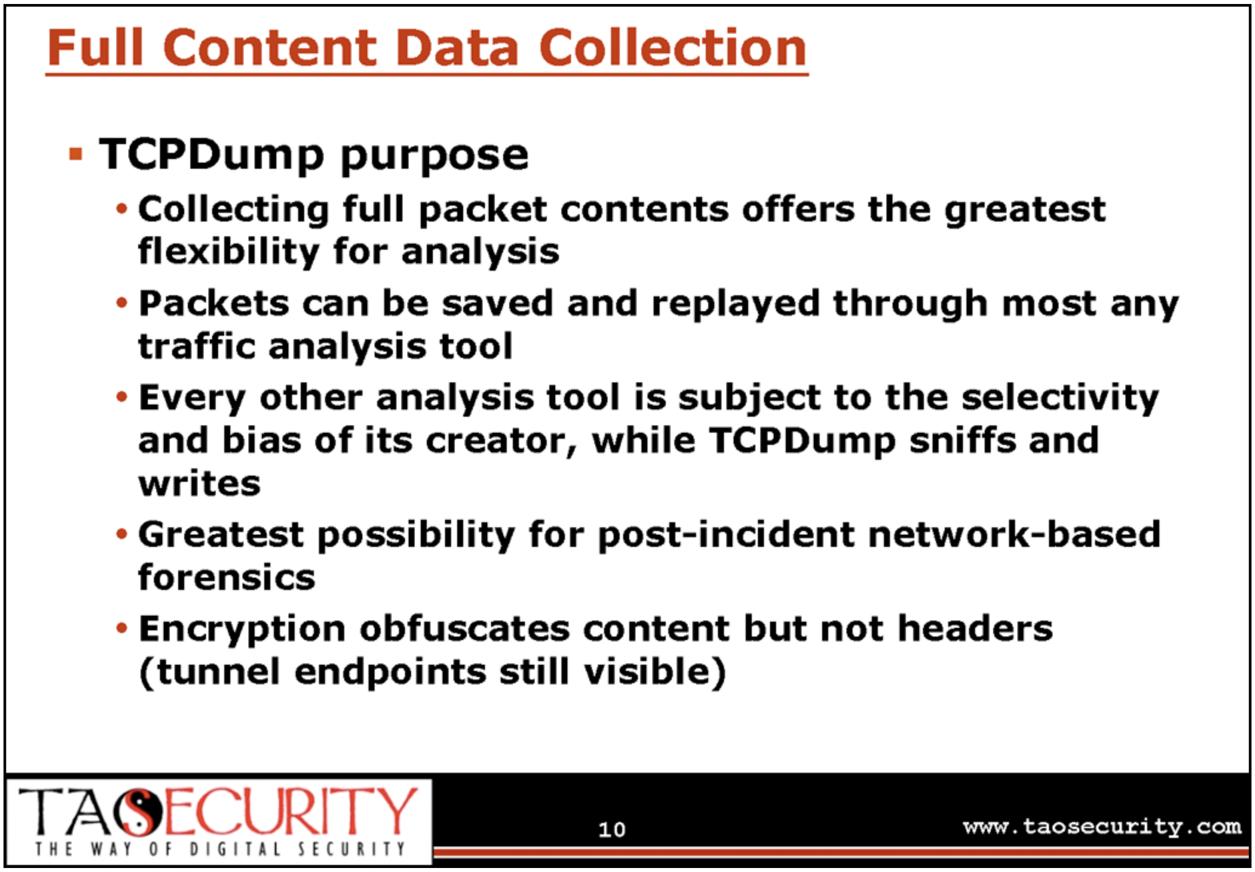
The better way, if possible, to get protection under another exception called the Consent Exception, meaning that the users of your site (which would presumably include the intruders) have consented to being monitored. If any of you have worked for the DOD, or any government departments, you probably remember sitting down at your Windows terminal or your Solaris box and seeing a big banner pop up with all these warnings about how you're basically signing your life away when you log in.

The same thing can be done on services that are bannerable. Although certain services certainly aren't bannerable. I mean, you can't banner an RPC service; you can't banner something that the user never sees. It is possible to banner Web sites, Telnet sessions, and things like that. This is a little more difficult, though, for obvious reasons.

I just cannot see the Department of Justice trying to prosecute someone who's been hacked, first of all, and prosecute that person because they were collecting traffic to try to defend his or her enterprise. I think that would cause such a backlash, it would be almost like attacking the victim, and I just don't see the DOJ doing something like that.

Could I have the next slide, please? OK, at this point we are at the slide that says "Data Collection Intro." We're now going to talk about the different tools we can use to collect data in a network security monitoring model. We're going to talk about T3 dump for full content data, Argus for session data, Snort for event data, trafd or trafshow for statistical data, and then we'll talk about Sguil as a package that is the closest to bringing more types of data into one box.

I know the title of the talk was implementing NSM with open source tools, but I will also give some commercial versions in case this is something that you'd be interested in. Something to keep in mind, though, is that the whole concept of NSM is not yet widely recognized in either the open source or the commercial world. So you're not going to find many products that are advertised as NSM tools. We've got a lot of other three-letter acronyms out there that are popular: IDS, ITS, SIM (Security Incident Management), etc., so you have to find tools that have the NSM stamp on them. But, the tools that I show you here (and once you start thinking about the types of data you can collect), you may find yourself looking at a tool and say, hey, I could use this to collect X type of data.



Full Content Data Collection

- TCPDump purpose
  - Collecting full packet contents offers the greatest flexibility for analysis
  - Packets can be saved and replayed through most any traffic analysis tool
  - Every other analysis tool is subject to the selectivity and bias of its creator, while TCPDump sniffs and writes
  - Greatest possibility for post-incident network-based forensics
  - Encryption obfuscates content but not headers (tunnel endpoints still visible)

TAOSECURITY THE WAY OF DIGITAL SECURITY 10 www.taosecurity.com

Figure 10

When I show you these tools and how to run them, or perhaps do different things with them, note that PowerPoint has a tendency to change quotes into crazy directions. Sometimes back ticks get turned into forward ticks and things like that. I do have screen shots that show you how to do certain activities, so in those cases, there will be more accuracy.

Could I have the next slide, please? So the first tool we'll talk about is TCPDump. TCPDump is pretty much the de facto full content data collection tool. If you're thinking that TCPDump is sort of an old tool, one that people use but there's really not a lot of development. It's pretty stable. I would invite you to go to tcpdump.org and take a look at any of the mailing lists. I don't subscribe to the mailing list any more because there's so much traffic on it. I read the archives instead. There is a lot of development going on in TCPDump and libpcap, the library which TCPDump relies upon. You'll find the same thing with certain other tools that are seen to be either stable or old code. So there's a lot going on in this area.

What's great about TCPDump, and any other tool that just collects traffic without any filtering, is that it offers you the greatest flexibility for analysis. What this means is that if you simply set the tool to run and collect traffic, you have a chance to catch that one-in-a-million intruder who is either backed by a foreign intelligence service or organized crime (or something to that effect) that is using the latest and greatest tool. A lot of the tools I'm going to talk about here can be beaten if you know how they work. But what I would submit to you is that if you run full collection, and the intruder has the ability to talk to your site, you will catch the intruder — or at least find evidence of that intruder's activities. Once we start moving away from collecting full content data and start talking about tools that aggregate, summarize, filter or alert data, you present an opportunity for the intruder to evade you. So, if at all possible, you should collect full content data with little to no filtering, so you have a chance to catch anybody that's out there. It may take a while; it may be weeks or months. This was the case when I was in the military and with certain other activities I've done. You may not know what you're looking at until a lot later. But if you've got that data, you can catch these guys.

What's also nice about full content data is you can pretty much replay it through any other tool. So if you have a tool that provides you with statistical data, you can feed that full content data through, perhaps collected at the T3 dump, and you'll get your statistics. Full content data also offers you the greatest capability for post-incident network based forensics. If you're simply grabbing everything and you don't know what to look for at the time, but later on you say: Oh, that's it — he's acting on port 12345. Then you've got that data collected.

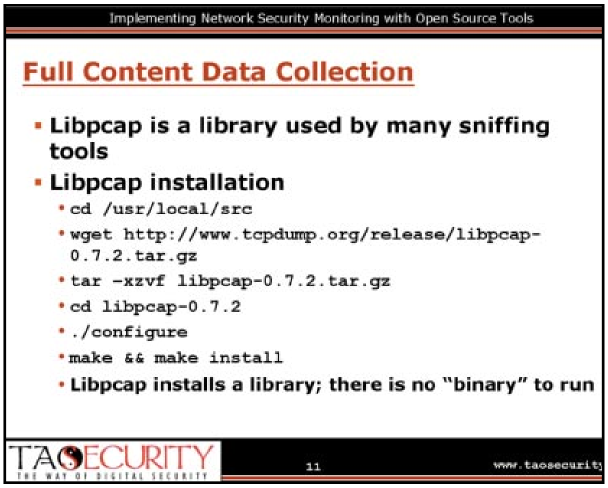
Here's something to keep in mind, too. People always say encryption will kill all sorts of intrusion detection and network security monitoring; it depends on what you're looking for. Sure, the encryption will obfuscate the content, but you'll also see other data that perhaps could be more important, such as where is the guy coming from, what time did he act and what machine he was speaking to. Then you can take other measures to find out what's going on.

Could I have the next slide, please? So in order to actually use TCPDump, you need to have a libpcap installed. Libpcap is the packet capture library upon which TCPDump drives for getting packets off the wire. I have some instructions here. Basically, if you go to tcpdump.org, you can download the latest and greatest. The instructions I give work fine on a stocked Redhead 7.3 box. I've said I'm a FreeBSD person. But for the sake of this demonstration I did everything on Redhead, since it's pretty popular.

Just remember, when you're done with installing the PCAP, there isn't binary. It's just a library, so once you're done, you're ready to go on and do other installations and tools.

Could I have the next slide, please? So you've got libpcap installed. Now you can take a look at TCPDump. I always recommend upgrading to the latest version of TCPDump and other tools, as well as the library in libpcap. There have been exploits in tools that simply sit and collect traffic off the wire. The most recent ones were exploits of Snort. It is conceivable that you could have a machine that is sitting, listening promiscuously on the wire, a specially formatted packet passes by and it overruns a buffer or takes some other action on your device that's running Snort or TCPDump, and causes that box to launch a shell outwards towards the victim's machine.

Now, it would have to go out of an interface that you would use for management. If you have a single interface device and it has no IP address, typically the exploit will fail. But there are vulnerabilities in older versions of products like Snort, TCPDump and TCPFlow that we won't talk about today, which have a vulnerability discovered recently. So I recommend that you follow the instructions to upgrade to the latest TCPDump. At the time of writing this presentation, it was 3.7.2.



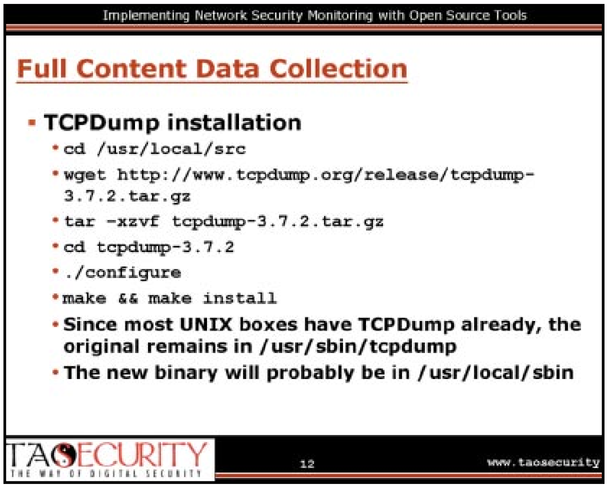
Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection

- Libpcap is a library used by many sniffing tools
- Libpcap installation
  - `cd /usr/local/src`
  - `wget http://www.tcpdump.org/release/libpcap-0.7.2.tar.gz`
  - `tar -xzvf libpcap-0.7.2.tar.gz`
  - `cd libpcap-0.7.2`
  - `./configure`
  - `make && make install`
  - Libpcap installs a library; there is no "binary" to run

TAOSECURITY THE WAY OF DIGITAL SECURITY 11 www.taosecurity

Figure 11



Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection

- TCPDump installation
  - `cd /usr/local/src`
  - `wget http://www.tcpdump.org/release/tcpdump-3.7.2.tar.gz`
  - `tar -xzvf tcpdump-3.7.2.tar.gz`
  - `cd tcpdump-3.7.2`
  - `./configure`
  - `make && make install`
  - Since most UNIX boxes have TCPDump already, the original remains in /usr/sbin/tcpdump
  - The new binary will probably be in /usr/local/sbin

TAOSECURITY THE WAY OF DIGITAL SECURITY 12 www.taosecurity

Figure 12

Could I have the next slide, please? So once you have libpcap and TCPDump installed, how do you run it? Well, there are certain switches that people use quite often. Some of the common ones I've got listed here. You specify the interface you want to watch the traffic with using –i. If don't want TCPDump to resolve IP address reports, you give it a –n. If you're running in test mode and you want to capture a certain number of packets, you can pass a -c with a number like 100 or 1,000, something like that. Probably the single most important switch here is the –s switch, which will tell you the size of the packet to capture. If you don't give it a size, it will only capture 68 bytes, so it's very important to specify –s. You want to tell the file to write traffic contents with a –w, and if you want to read, you can use a –r. If you're reading the traffic back, you can use –tttt to show the date and time stamp. And to see the packet contents, you can use a –X (capital X).

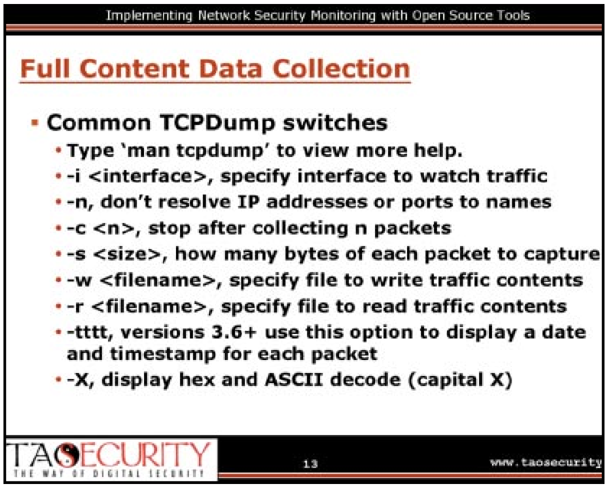
May I have the next slide, please? Here are a couple of typical usage statements. We won't go through them, but basically we tell TCPDump to capture some traffic, and then if you want to read it back in, you can do so.

When reviewing raw TCPDump data, most people prefer to use something like ethereal to take a look at the data. I agree that this is a good idea. Unless you're looking through a huge amount of data … if you've limited your view or you know a certain combination of IPs or ports are of interest — then you can use ethereal. There are things called Berkeley Packet Filters, a syntax you can use to modify TCPDump's behavior either on the front end when you're collecting the traffic or on the backend when you're analyzing the traffic. So you could tell TCPDump to look at host 10.1.1.1 and port 80, for example.

Now, I'd like to discuss just a couple of notes on actually collecting the traffic. I always recommend that when you write traffic captures to disk that you use a naming convention based on the time and date that the capture started. Also, if you can, add in the host name and the interface where the traffic was collected. It's also a good idea whenever you're collecting traffic to do it on a dedicated partition. I always create a /nsm partition. If for some reason the scripts that you use to control traffic collection go bad, or you don't remember to activate them for whatever reason, if you fill up that /nsm partition, you won't crash the rest of the box. It's a nasty thing when you fill up the repartition on a Unix box. So it's always good to send your traffic to a dedicated partition.

Can I have the next slide, please? What we have here is just a sample of your TCPDump output.

Next slide, please. So if you're thinking it's good idea to collect all this traffic for the flexibility and the information it provides, but you want a commercial solution, the two that are most well-known out there in the world are SandStorm NetIntercept and the Niksun NetDetector. Again, let me point out that I'm not paid by anybody; I don't have any of these boxes. Of the two that I've seen, I tend to like the NetIntercept. I like the interface a



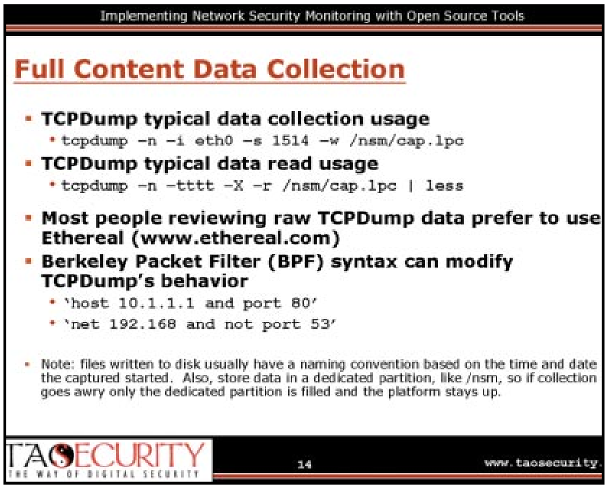
Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection

- Common TCPDump switches
  - Type 'man tcpdump' to view more help.
  - -i <interface>, specify interface to watch traffic
  - -n, don't resolve IP addresses or ports to names
  - -c <n>, stop after collecting n packets
  - -s <size>, how many bytes of each packet to capture
  - -w <filename>, specify file to write traffic contents
  - -r <filename>, specify file to read traffic contents
  - -tttt, versions 3.6+ use this option to display a date and timestamp for each packet
  - -X, display hex and ASCII decode (capital X)

TAOSECURITY THE WAY OF DIGITAL SECURITY 13 www.taosecurity

Figure 13



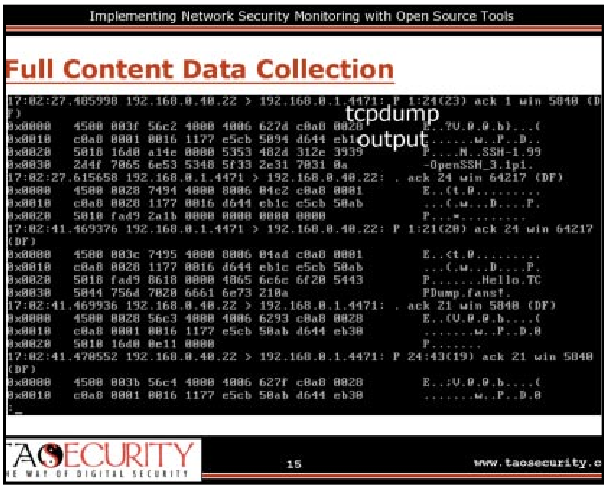
Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection

- TCPDump typical data collection usage
  - tcpdump -n -i eth0 -s 1514 -w /nsm/cap.lpc
- TCPDump typical data read usage
  - tcpdump -n -tttt -X -r /nsm/cap.lpc | less
- Most people reviewing raw TCPDump data prefer to use Ethereal (www.ethereal.com)
- Berkeley Packet Filter (BPF) syntax can modify TCPDump's behavior
  - 'host 10.1.1.1 and port 80'
  - 'net 192.168 and not port 53'
- Note: files written to disk usually have a naming convention based on the time and date the captured started. Also, store data in a dedicated partition, like /nsm, so if collection goes awry only the dedicated partition is filled and the platform stays up.

TAOSECURITY THE WAY OF DIGITAL SECURITY 14 www.taosecurity.

Figure 14



Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection

```
17:02:27.485998 192.168.0.40.22 > 192.168.0.1.4471: P 1:24(23) ack 1 win 5840 (D
F)
0x0000   4500 003f 56c2 4000 4006 627d c0a8 0028         E..?V.@.@.b}...(
0x0010   c0a8 0001 0016 1177 e5cb 5094 d644 eb1c         ........w..P..D..
0x0020   5018 16d0 a14e 0000 5353 482d 312e 3939         P....N..SSH-1.99
0x0030   2d4f 7065 6e53 5348 5f33 2e31 7031 0a           -OpenSSH_3.1p1.
17:02:27.615658 192.168.0.1.4471 > 192.168.0.40.22: . ack 24 win 64217 (DF)
0x0000   4500 0028 7494 4000 8006 04c2 c0a8 0001         E..(t.@.........
0x0010   c0a8 0028 1177 0016 d644 eb1c e5cb 50ab         ...(.w...D....P.
0x0020   5018 fad9 2a1b 0000 0000 0000 0000              P...*.........
17:02:41.469376 192.168.0.1.4471 > 192.168.0.40.22: P 1:21(20) ack 24 win 64217
(DF)
0x0000   4500 003c 7495 4000 8006 04ad c0a8 0001         E..<t.@.........
0x0010   c0a8 0028 1177 0016 d644 eb1c e5cb 50ab         ...(.w...D....P.
0x0020   5018 fad9 0618 0000 4865 6c6c 6f20 5443         P.......Hello.TC
0x0030   5044 756d 7020 6661 6e73 210a                   PDump.fans!.
17:02:41.469936 192.168.0.40.22 > 192.168.0.1.4471: . ack 21 win 5840 (DF)
0x0000   4500 0028 56c3 4000 4006 6293 c0a8 0028         E..(V.@.@.b....(
0x0010   c0a8 0001 0016 1177 e5cb 50ab d644 eb30         ........w..P..D.0
0x0020   5010 16d0 0e11 0000                             P.......
17:02:41.470552 192.168.0.40.22 > 192.168.0.1.4471: P 24:43(19) ack 21 win 5840
(DF)
0x0000   4500 003b 56c4 4000 4006 627f c0a8 0028         E..;V.@.@.b....(
0x0010   c0a8 0001 0016 1177 e5cb 50ab d644 eb30         ........w..P..D.0
:_
```

tcpdump output

TAOSECURITY THE WAY OF DIGITAL SECURITY 15 www.taosecurity.c

Figure 15

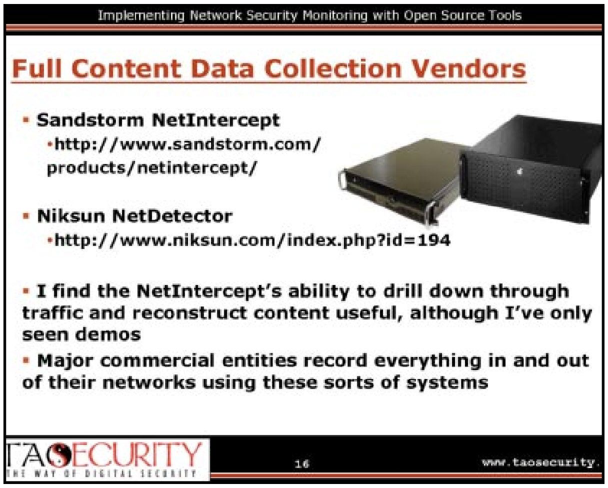
little bit better and it seems to be more oriented towards the type of network-based forensics that I'm advocating.

I know of major commercial entities, banks, insurance companies, etc., that use one or more of these boxes to grab everything that's going in and out of their networks, and do to that, provide that network forensic or auditing capability.

Next slide, please. So we've talked about the full content data collection. Now let's move on to the session data collection. The tool that I like to use is called Argus. It's been around since 1995 and it was invented by a gentleman at Carnegie Mellon University called Carter Boyd. Argus is great because it offers you the ability to summarize traffic — IP, TCP, UDP, ICMP traffic — in what you might call a conversation or session format. It saves the data in a proprietary format without storing headers and parses it on the backend. What's great about this is that the format it stores it in is so compact; in some cases, you could store months to years worth of Argus data on a relatively small hard drive. Argus is also great because it's not fooled by encryption. It sorts out these source ports, IP destination ports, packets and bytes that were sent in either direction. It doesn't care what the content is. So Argus is great because it keeps track of who talked to whom and when. The way to beat Argus is to use some sort of tool that's completely sessionless; Argus has a hard time keeping track of that.

Next slide, please. If you want to install Argus, you can download it at www.qosient.com/argus. The version that's listed there as being publicly available is 2.0.5. Again, Argus is one of these tools where if you go to the Web site, it doesn't look like a whole lot is going on. But if you go to the mailing list, there's a lot of development going on. I've actually talked to Carter recently about trying to get a little better public relations base on the Argus site so that people will know that this great tool is undergoing active development. Carter is incredibly responsive to user input, and if you have any problems, he's very willing to help. So if you do have an interest in Argus, I recommend you join the mailing list or send an e-mail to Carter.

Next slide, please. Keep in mind that Argus is two pieces: there's a server and a client. The server piece is the part that collects the traffic off the wire, and the client piece is the part that you use to read the traffic that was collected. This is not collecting libpcap formatted data. In other words, if you use

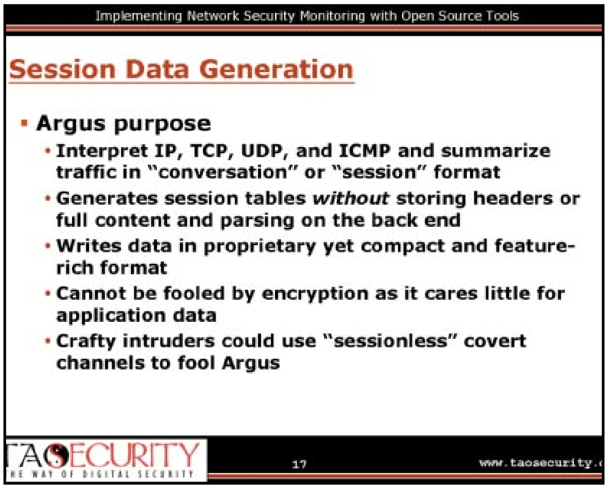

Implementing Network Security Monitoring with Open Source Tools

Full Content Data Collection Vendors

- Sandstorm NetIntercept
  - http://www.sandstorm.com/products/netintercept/
- Niksun NetDetector
  - http://www.niksun.com/index.php?id=194
- I find the NetIntercept's ability to drill down through traffic and reconstruct content useful, although I've only seen demos
- Major commercial entities record everything in and out of their networks using these sorts of systems

TAOSECURITY THE WAY OF DIGITAL SECURITY 16 www.taosecurity.

Figure 16

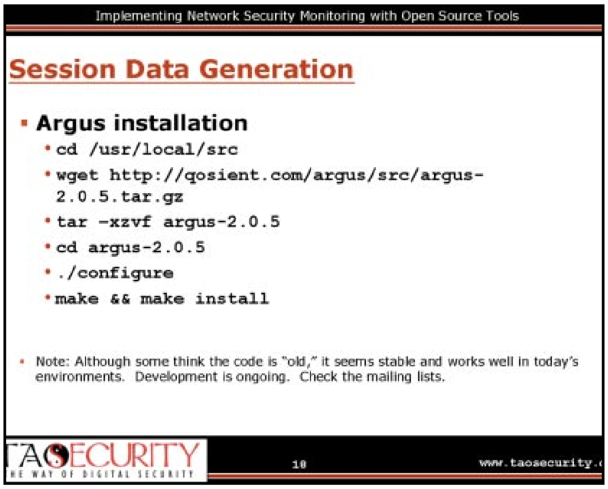

Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Argus purpose
  - Interpret IP, TCP, UDP, and ICMP and summarize traffic in "conversation" or "session" format
  - Generates session tables *without* storing headers or full content and parsing on the back end
  - Writes data in proprietary yet compact and feature-rich format
  - Cannot be fooled by encryption as it cares little for application data
  - Crafty intruders could use "sessionless" covert channels to fool Argus

TAOSECURITY THE WAY OF DIGITAL SECURITY 17 www.taosecurity.

Figure 17


Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Argus installation
  - `cd /usr/local/src`
  - `wget http://qosient.com/argus/src/argus-2.0.5.tar.gz`
  - `tar -xzvf argus-2.0.5`
  - `cd argus-2.0.5`
  - `./configure`
  - `make && make install`
- Note: Although some think the code is "old," it seems stable and works well in today's environments. Development is ongoing. Check the mailing lists.

TAOSECURITY THE WAY OF DIGITAL SECURITY 18 www.taosecurity.

Figure 18

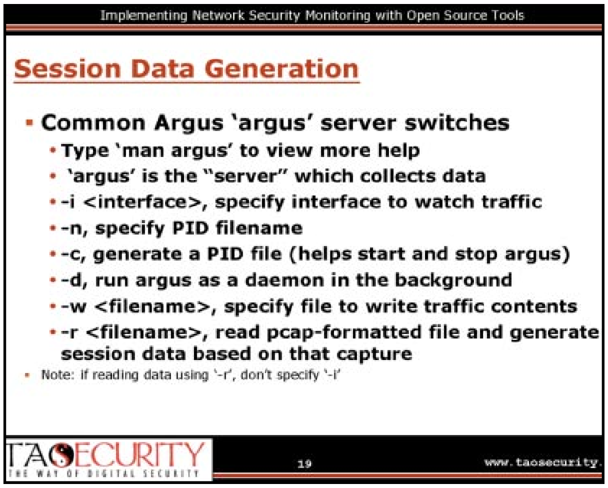
Argus and you write the traffic to a file, you cannot read it using TCPDump. You cannot read it using a serial. Argus writes its data in its own format and only the RA client can read that data. Now, there are a lot of different options you can pass to RA in order to format the data in various means — but again, Argus is a single-purpose tool. It is trying to watch a lot of traffic, summarize it and write what it sees into tables.

This is something that is a little bit different from other tools. Other tools will generate session data. But the way they generally do it is by grabbing all of the data and storing it to disk. And, on the backend, in a batch mode, they park that data generally to sessions. Argus is nice because it does it all in memory. It doesn't write anything to disk in terms of headers or content or whatever. It sees traffic, it builds tables in memory and it writes those tables to the disk as it sees it. So I've got some usage here in terms of flags to pass.

Could I have the next slide, please? On this slide I've got two options of how you could use Argus. The first is an example of using Argus in a live mode; this is the way I do it. I simply start it up, run it against an interface and tell it to write what it sees into a file in the NSM partition. If you've already collected traffic and you'd like to see what Argus makes of it — say you've caught the traffic with TCPDump — you can run Argus against that TCPDump capture file and then Argus will produce its own Argus-formatted file. So you'd have two files — your libpcap file and your Argus file.

Could I have the next slide, please? Once you've Argus to write the traffic to its proprietary format on the hard drive, you use the RA client (and I imagine that stands for read Argus) to take a look at that traffic and figure out what's going on. I've got the syntax here, the different options that you can use. The thing that's nice about Argus is you can see as little or as much as you want of Argus's capabilities. In its very, very basic mode it will simply say, I saw a source IP, destination IP, source port, destination port and maybe some packet counts. If you want, though, you can tell Argus to show you the states of the connections, meaning whether the connection was simply a SIM, if it completed, if a three-way handshake completed, if the session completed gracefully, and whether the session ended with a reset.

I've been talking a lot in terms of TCP, but Argus will also make its best guess as to what it sees with UDP. So if there's a DNS request and there's a reply,



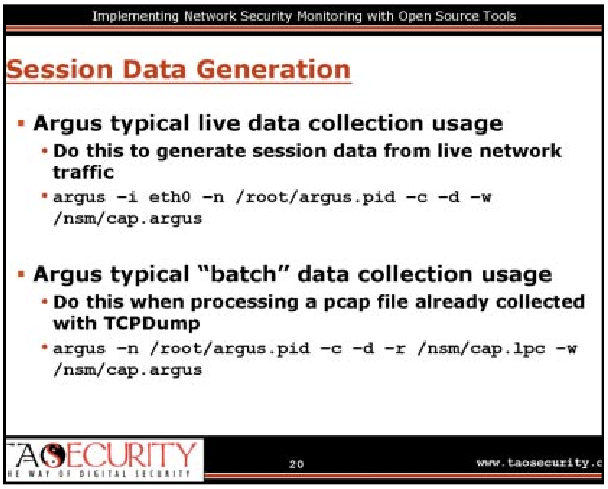
Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Common Argus 'argus' server switches
  - Type 'man argus' to view more help
  - 'argus' is the "server" which collects data
  - -i <interface>, specify interface to watch traffic
  - -n, specify PID filename
  - -c, generate a PID file (helps start and stop argus)
  - -d, run argus as a daemon in the background
  - -w <filename>, specify file to write traffic contents
  - -r <filename>, read pcap-formatted file and generate session data based on that capture
- Note: if reading data using '-r', don't specify '-i'

TAOSECURITY THE WAY OF DIGITAL SECURITY 19 www.taosecurity.

Figure 19



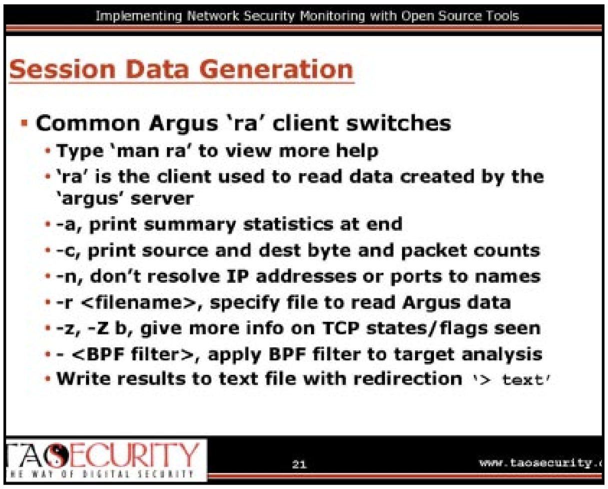
Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Argus typical live data collection usage
  - Do this to generate session data from live network traffic
  - `argus -i eth0 -n /root/argus.pid -c -d -w /nsm/cap.argus`
- Argus typical "batch" data collection usage
  - Do this when processing a pcap file already collected with TCPDump
  - `argus -n /root/argus.pid -c -d -r /nsm/cap.lpc -w /nsm/cap.argus`

TAOSECURITY HE WAY OF DIGITAL SECURITY 20 www.taosecurity.c

Figure 20



Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Common Argus 'ra' client switches
  - Type 'man ra' to view more help
  - 'ra' is the client used to read data created by the 'argus' server
  - -a, print summary statistics at end
  - -c, print source and dest byte and packet counts
  - -n, don't resolve IP addresses or ports to names
  - -r <filename>, specify file to read Argus data
  - -z, -Z b, give more info on TCP states/flags seen
  - - <BPF filter>, apply BPF filter to target analysis
  - Write results to text file with redirection '> text'

TAOSECURITY HE WAY OF DIGITAL SECURITY 21 www.taosecurity.c

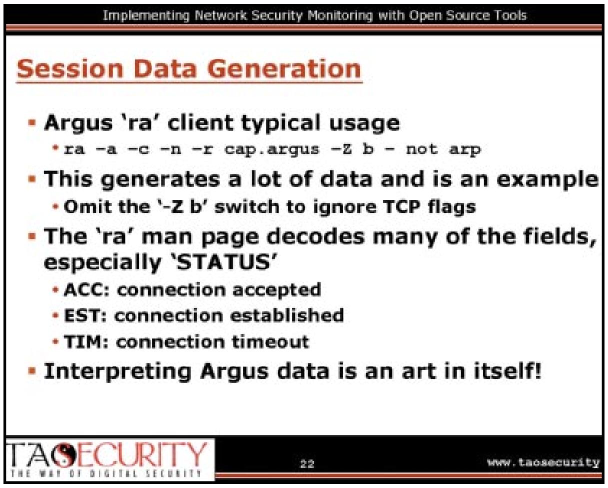
Figure 21

Argus will try to treat that as a session. It will also keep track of ICNP traffic — it will make sure certain traffic is request response. So you can almost consider that a session. Of course, I'm starting to blur boundaries here, but Argus is pretty amazing once you start using it.

Could I have the next slide, please? Here's an example of using the RA client; I'm simply reading in an Argus file called cap.argus. The interesting thing about Argus is that the more you start to get into it and you start to look at its other features, it's almost an art unto itself how to interpret some of this data. For example, you could take a look at Argus data and say, I want to see the flags that were seen during the conversation. Well, if you seeing an acroset, what does that mean? Well, it could mean that someone scanned your network; they hit you with a SIM packet. Your box replied with a Synac. Right? Or it could mean that someone hit your box, they hit you with a Synac, and there was nothing else. So you have to start looking at the flags that were set, maybe the number of packets that were sent, and already you can see if you were collecting full content data; you would be able to go back and look at the raw traffic and see exactly what happened. But because you're moving a step up, trying to save disk space and keep traffic statistics or traffic data on a loaded network, you're starting to use Argus; you're losing a little bit of the fidelity, but it's better than nothing.

Next slide, please. This slide shows you an example of Argus output and has a couple of simple connections, one to TCP and a couple of ICMP packets that were sent.

Could I have the next slide, please? In terms of commercial products, there really isn't a whole lot of activity in this field. I think there will be more now that there are starting to become guidelines for auditing network traffic. The one product that I think really seems to understand this is called StealthWatch, which is created by a company called Lancope. The products I mentioned earlier, NetIntercept and NetDetector, do offer a certain level of session data, but they only do so by collecting all the traffic, or as much traffic as you tell them to collect, and then parsing it on the backend. So keep in mind that tools like StealthWatch and Argus will collect traffic without writing it all to disk. They'll keep the traffic in memory and then summarize it.



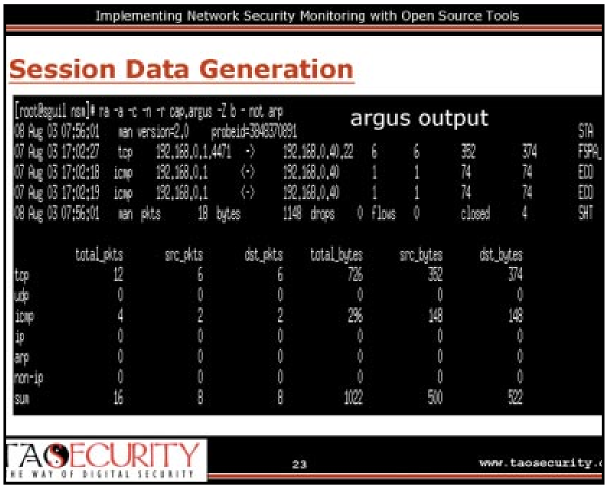
Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

- Argus 'ra' client typical usage
  - ra -a -c -n -r cap.argus -Z b - not arp
- This generates a lot of data and is an example
  - Omit the '-Z b' switch to ignore TCP flags
- The 'ra' man page decodes many of the fields, especially 'STATUS'
  - ACC: connection accepted
  - EST: connection established
  - TIM: connection timeout
- Interpreting Argus data is an art in itself!

TAOSECURITY THE WAY OF DIGITAL SECURITY 22 www.taosecurity

Figure 22



Implementing Network Security Monitoring with Open Source Tools

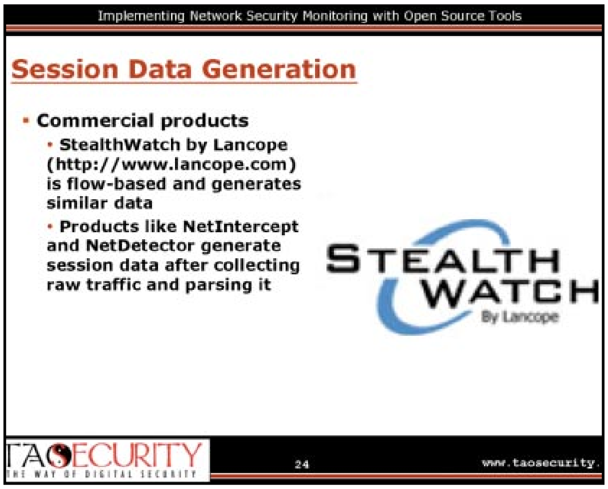
Session Data Generation

argus output

```
[root@sguil nsm]# ra -a -c -n -r cap.argus -Z b - not arp
08 Aug 03 07:56:01    man version=2.0    probeid=3848370891                                   STA
07 Aug 03 17:02:27    tcp    192.168.0.1.4471   ->    192.168.0.40.22   6     6     352     374     FSPA_
07 Aug 03 17:02:18    icmp   192.168.0.1        <->   192.168.0.40      1     1     74      74      ECO
07 Aug 03 17:02:19    icmp   192.168.0.1        <->   192.168.0.40      1     1     74      74      ECO
08 Aug 03 07:56:01    man  pkts        18  bytes       1148  drops    0 flows   0     closed   4      SHT

          total_pkts      src_pkts      dst_pkts   total_bytes     src_bytes     dst_bytes
tcp               12             6             6           726           352           374
udp                0             0             0             0             0             0
icmp               4             2             2           296           148           148
ip                 0             0             0             0             0             0
arp                0             0             0             0             0             0
non-ip             0             0             0             0             0             0
sum               16             8             8          1022           500           522
```

TAOSECURITY THE WAY OF DIGITAL SECURITY 23 www.taosecurity.c

Figure 23



Implementing Network Security Monitoring with Open Source Tools

Session Data Generation

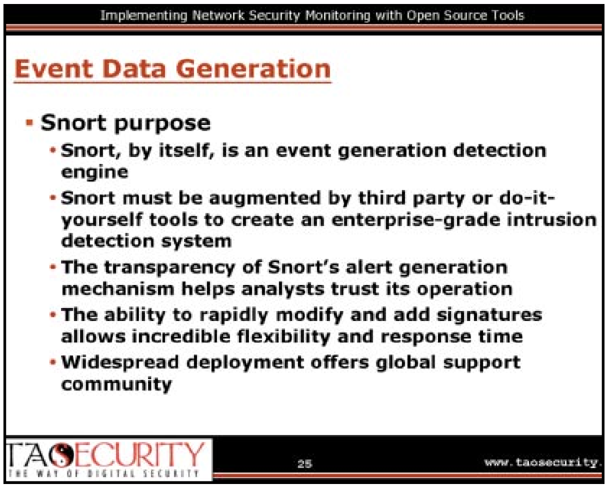
- Commercial products
  - StealthWatch by Lancope (http://www.lancope.com) is flow-based and generates similar data
  - Products like NetIntercept and NetDetector generate session data after collecting raw traffic and parsing it

STEALTH WATCH
By Lancope

TAOSECURITY THE WAY OF DIGITAL SECURITY 24 www.taosecurity.

Figure 24

Next slide, please. You can't talk about security monitoring or IDF without mentioning Snort, so I'll just quickly give you a little bit on Snort. I view Snort as being an event detection engine. You've got to augment it with third-party or do-it-yourself tools to really create an enterprise-grade intrusion detection system. The great thing about Snort, though, is that it's so transparent. You can see how it works, read the code and signatures and make your own signatures. You build trust in that system because you know how it works. You can rapidly modify it. You can add signatures. And, literally within a minute of something new that is attacking the whole Internet, you can go to the Snort-users Web site and someone has posted a signature. It may not be the best signature in the world, but you know someone else will post another one and pretty soon you've got a community consensus signature for whatever activity is out there. So Snort is great for that purpose.



Implementing Network Security Monitoring with Open Source Tools

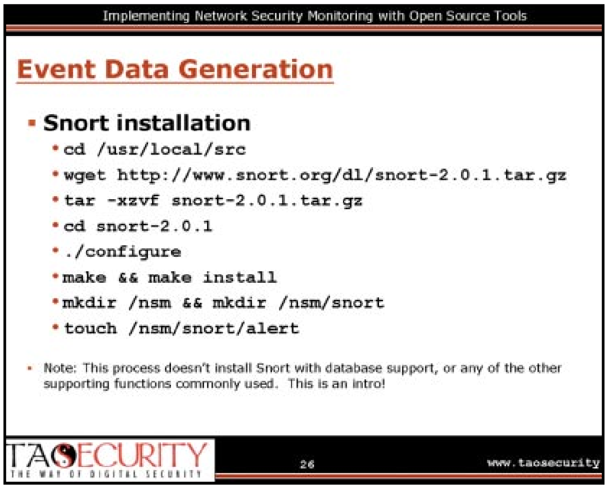
Event Data Generation

- Snort purpose
  - Snort, by itself, is an event generation detection engine
  - Snort must be augmented by third party or do-it-yourself tools to create an enterprise-grade intrusion detection system
  - The transparency of Snort's alert generation mechanism helps analysts trust its operation
  - The ability to rapidly modify and add signatures allows incredible flexibility and response time
  - Widespread deployment offers global support community

TAOSECURITY THE WAY OF DIGITAL SECURITY 25 www.taosecurity.

Figure 25

Next slide, please. If you've never installed Snort, I've got half a dozen instructions here on how to do it. It's very simple. Snort is another product which has had vulnerabilities recently. So I always recommend downloading the latest version. As of writing these slides, the latest version was the 2.0.1.



Implementing Network Security Monitoring with Open Source Tools

Event Data Generation

- Snort installation
  - `cd /usr/local/src`
  - `wget http://www.snort.org/dl/snort-2.0.1.tar.gz`
  - `tar -xzvf snort-2.0.1.tar.gz`
  - `cd snort-2.0.1`
  - `./configure`
  - `make && make install`
  - `mkdir /nsm && mkdir /nsm/snort`
  - `touch /nsm/snort/alert`
- Note: This process doesn't install Snort with database support, or any of the other supporting functions commonly used. This is an intro!

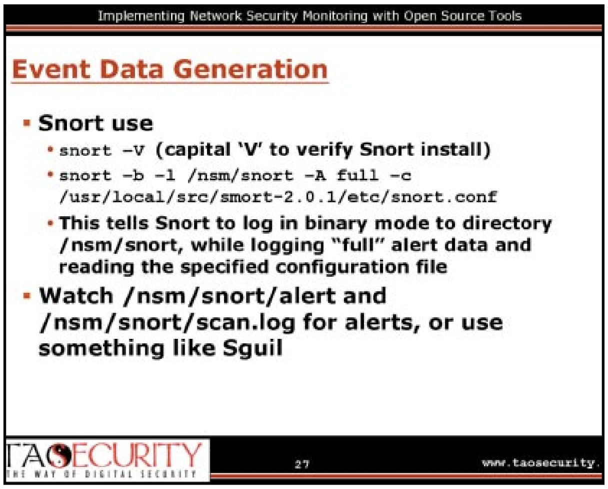
TAOSECURITY THE WAY OF DIGITAL SECURITY 26 www.taosecurity

Figure 26

Could I have the next slide, please? Once you've run through the instructions, test your Snort install by doing a snort with a –V (capital V). If you get no errors, then Snort is running. I've got some syntax on the slide here for running Snort in a full alert data mode. Keep in mind that this will simply write the output into a file called snort/alert if you've followed the instructions, and they create a file called scan.log for alerts. This isn't sufficient for an enterprise, right? So you need to move on to something else; we'll talk about that shortly.

Could I have the next slide, please? This slide shows you a raw example of what text-based Snort alerts look like.

Next slide, please. So in terms of the commercial side, this is where the vendors have products. There's been a lot of attention in this area, and has been probably for the last five years, ever since ISS came out with the first version. Real secure, the wheel grid, created a net range or things like that. Of the commercial IDSes out there, if I'm touting Snort, I'm obviously going to speak about Sourcefire. Sourcefire is the commercial version of Snort. It takes the Snort engine, the same one you can download for free that everybody uses, and it



Implementing Network Security Monitoring with Open Source Tools

Event Data Generation

- Snort use
  - `snort –V` (capital 'V' to verify Snort install)
  - `snort -b -l /nsm/snort -A full -c /usr/local/src/smort-2.0.1/etc/snort.conf`
  - This tells Snort to log in binary mode to directory /nsm/snort, while logging "full" alert data and reading the specified configuration file
- Watch /nsm/snort/alert and /nsm/snort/scan.log for alerts, or use something like Sguil

TAOSECURITY THE WAY OF DIGITAL SECURITY 27 www.taosecurity.

Figure 27

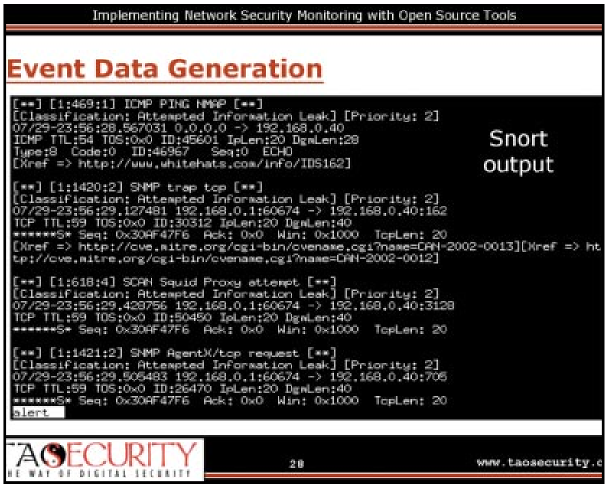
packages it with all of the different enterprise-grade products you need to run Snort in hierarchical reporting, database event management style. Again, I don’t sell Snort and I don’t sell Sourcefire. I just think it’s the best that’s out there. Of the commercial IDS products that predated Sourcefire, I think the one that was most congruent with NSM principles was Dragon, which is now built by Enterasys.

Could I have the next slide, please? We’re almost done winding up the tools discussion. Now let's talk about statistical data. Statistical data is what you normally think about when doing network performance monitoring or health monitoring, rather than NSM. The reason why I mention it here is that if you don’t control the routers, or you don’t control the firewalls in your organization, but you’d sort of like to know what’s the level of traffic going through those devices or near those devices, you can collect certain types of data with products like trafd or trafshow. For example, trafd will show you statistics that are similar to Cisco accounting data. It collects all this traffic in memory. So it’s like the top talkers -- what is the most active or reports the most active and so forth. You can then dump that to disk periodically. If you want to see it in a real-time mode — what’s happening right now — you can use trafshow.

I used trafshow. A customer would call me and say his bandwidth was terrible and would want to know what was going on. I would log into the sensor, file trafshow and see that there was a huge peer-to-peer session going on. Someone was downloading the latest version of Linux or a movie or something like that. I like these because you can secure (inaudible) in the sensor and look at both of them in a text terminal.

I tried to install these onto my Redhead 7.3. Actually I tried to install trafd, but it didn’t compile clean. If you’re using FreeBSD, you can install both of these products from their respective ports in user ports net. I was able to install trafshow without a problem on my Redhead 7.3 test box.

Could I have the next slide, please? You should be looking at the slide that says trafd use and trafshow use. Both of these are pretty simple to use. You simply fire them up and let them run against a certain interface. With trafd, it’s a game and it sits in the background and you view the traffic using trafstat. With trafshow, because it’s a real-time tool, you simply run trafshow, and it will show you the traffic it sees in real time.



Implementing Network Security Monitoring with Open Source Tools

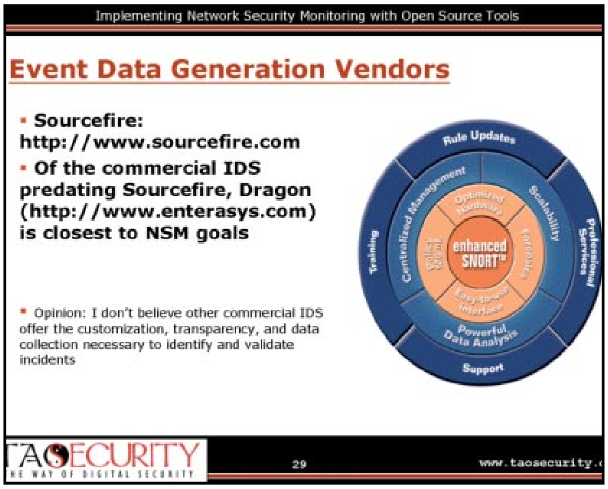
Event Data Generation

```
[**] [1:469:1] ICMP PING NMAP [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/29-23:56:28.567031 0.0.0.0 -> 192.168.0.40
ICMP TTL:54 TOS:0x0 ID:45601 IpLen:20 DgmLen:28
Type:8  Code:0  ID:46967   Seq:0  ECHO
[Xref => http://www.whitehats.com/info/IDS162]

[**] [1:1420:2] SNMP trap tcp [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/29-23:56:29.127481 192.168.0.1:60674 -> 192.168.0.40:162
TCP TTL:59 TOS:0x0 ID:30312 IpLen:20 DgmLen:40
******S* Seq: 0x30AF47F6  Ack: 0x0  Win: 0x1000  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0013][Xref => ht
tp://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0012]

[**] [1:618:4] SCAN Squid Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/29-23:56:29.428756 192.168.0.1:60674 -> 192.168.0.40:3128
TCP TTL:59 TOS:0x0 ID:50450 IpLen:20 DgmLen:40
******S* Seq: 0x30AF47F6  Ack: 0x0  Win: 0x1000  TcpLen: 20

[**] [1:1421:2] SNMP AgentX/tcp request [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/29-23:56:29.506483 192.168.0.1:60674 -> 192.168.0.40:705
TCP TTL:59 TOS:0x0 ID:26470 IpLen:20 DgmLen:40
******S* Seq: 0x30AF47F6  Ack: 0x0  Win: 0x1000  TcpLen: 20
alert
```

Snort output

TAOSECURITY THE WAY OF DIGITAL SECURITY 28 www.taosecurity.c

Figure 28



Implementing Network Security Monitoring with Open Source Tools

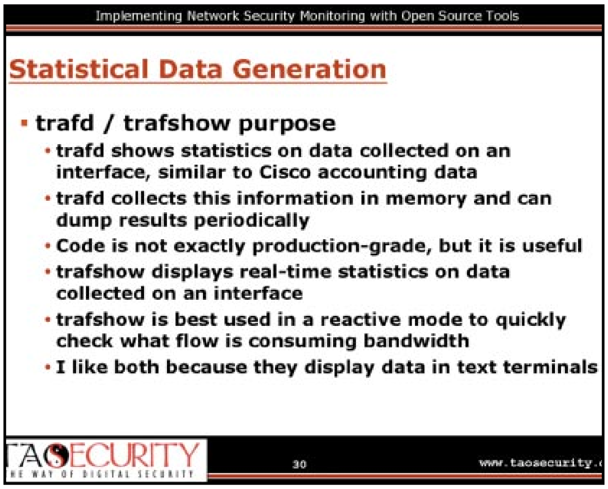
Event Data Generation Vendors

- Sourcefire: http://www.sourcefire.com
- Of the commercial IDS predating Sourcefire, Dragon (http://www.enterasys.com) is closest to NSM goals

- Opinion: I don’t believe other commercial IDS offer the customization, transparency, and data collection necessary to identify and validate incidents

TAOSECURITY THE WAY OF DIGITAL SECURITY 29 www.taosecurity.c

Figure 29



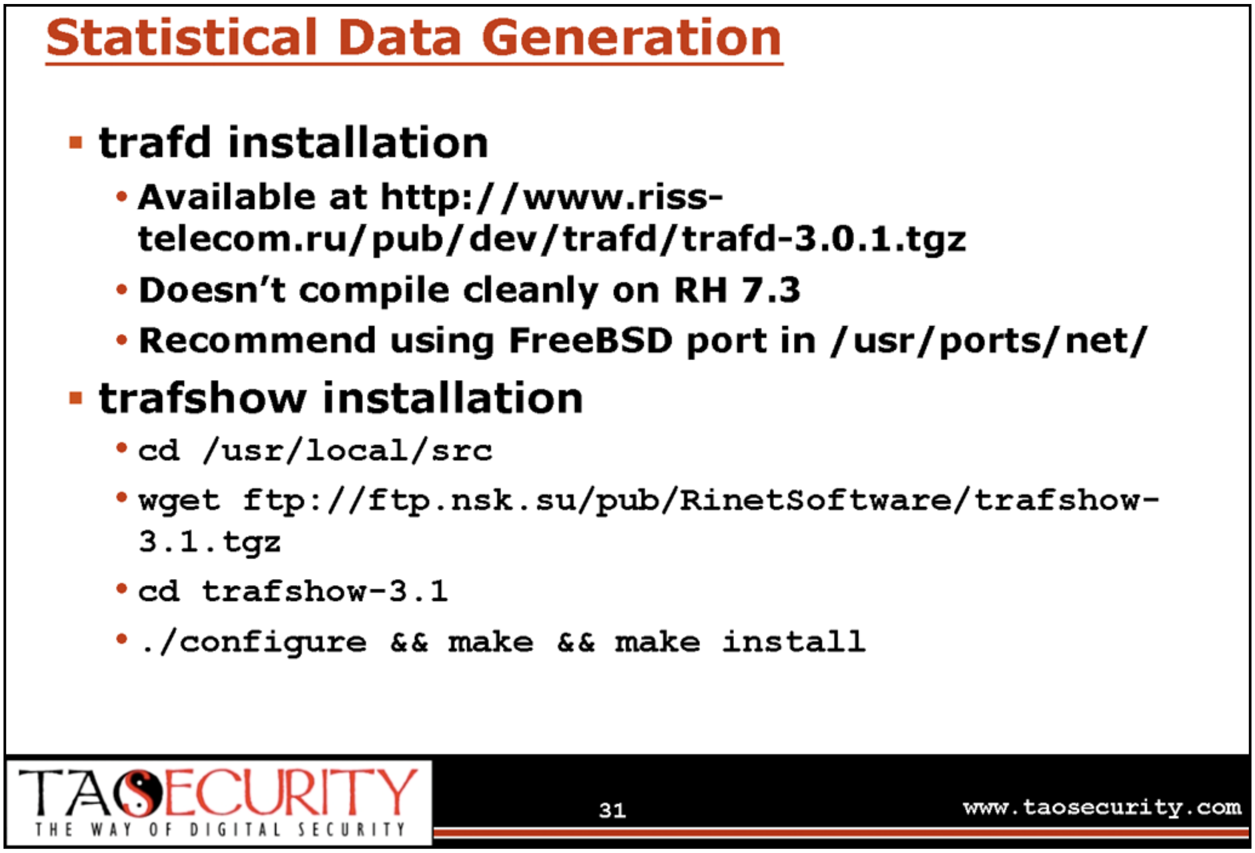
Implementing Network Security Monitoring with Open Source Tools

Statistical Data Generation

- trafd / trafshow purpose
  - trafd shows statistics on data collected on an interface, similar to Cisco accounting data
  - trafd collects this information in memory and can dump results periodically
  - Code is not exactly production-grade, but it is useful
  - trafshow displays real-time statistics on data collected on an interface
  - trafshow is best used in a reactive mode to quickly check what flow is consuming bandwidth
  - I like both because they display data in text terminals

TAOSECURITY THE WAY OF DIGITAL SECURITY 30 www.taosecurity.c

Figure 30

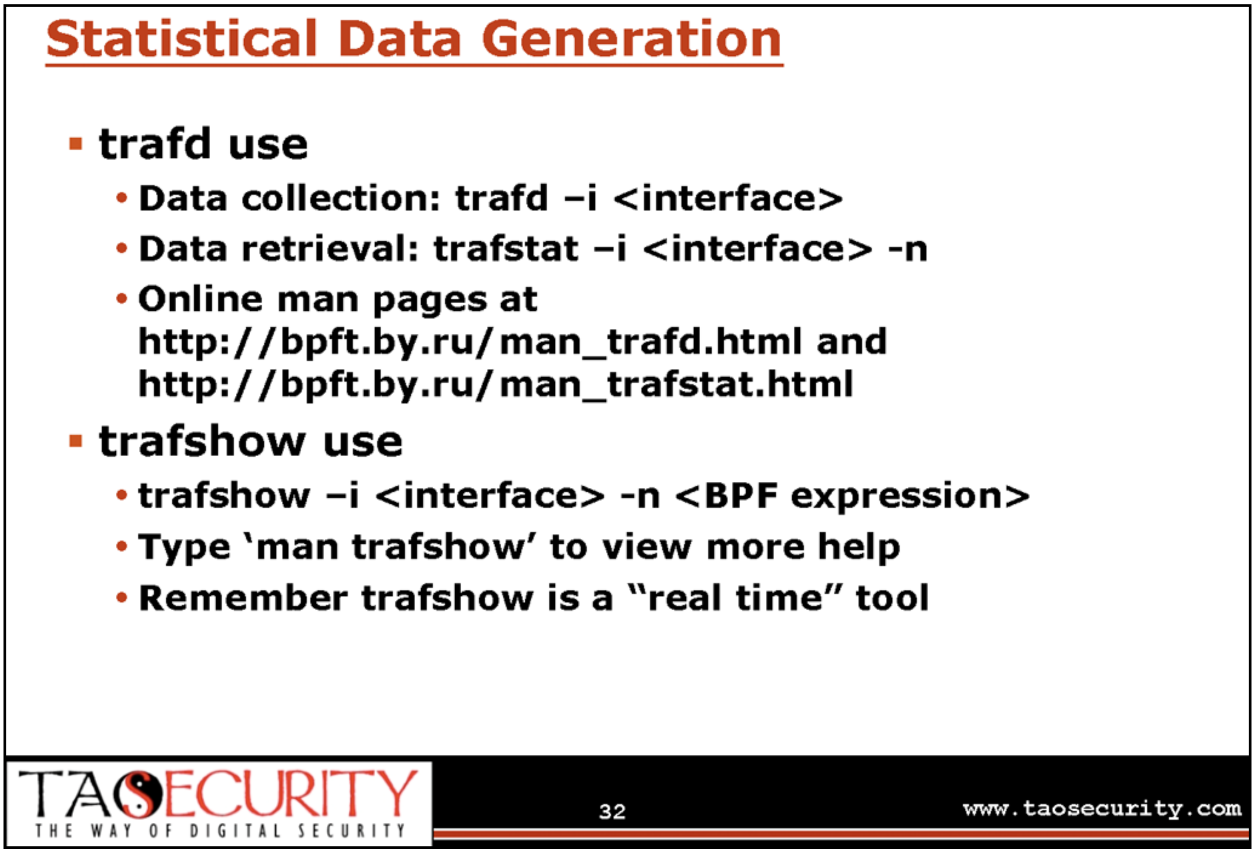


Statistical Data Generation

- trafd installation
  - Available at http://www.riss-telecom.ru/pub/dev/trafd/trafd-3.0.1.tgz
  - Doesn't compile cleanly on RH 7.3
  - Recommend using FreeBSD port in /usr/ports/net/
- trafshow installation
  - `cd /usr/local/src`
  - `wget ftp://ftp.nsk.su/pub/RinetSoftware/trafshow-3.1.tgz`
  - `cd trafshow-3.1`
  - `./configure && make && make install`

TAOSECURITY THE WAY OF DIGITAL SECURITY 31 www.taosecurity.com

Figure 31



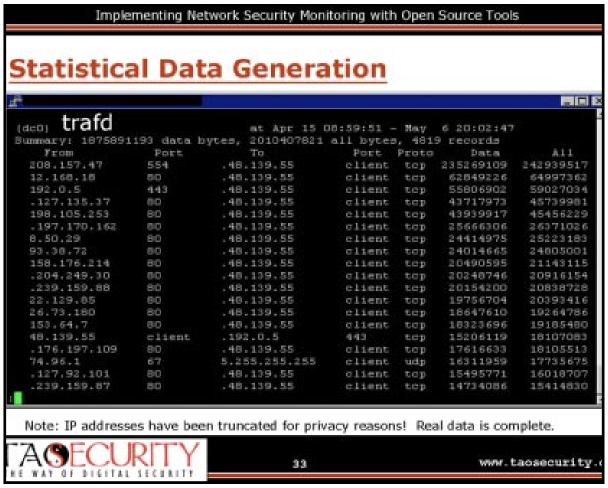
Statistical Data Generation

- trafd use
  - Data collection: trafd –i <interface>
  - Data retrieval: trafstat –i <interface> -n
  - Online man pages at http://bpft.by.ru/man_trafd.html and http://bpft.by.ru/man_trafstat.html
- trafshow use
  - trafshow –i <interface> -n <BPF expression>
  - Type 'man trafshow' to view more help
  - Remember trafshow is a "real time" tool

TAOSECURITY THE WAY OF DIGITAL SECURITY 32 www.taosecurity.com

Figure 32

Could I have the next slide, please? I've got a couple of screen captures here. The first screen capture is trafd. I've edited the traffic here and truncated the first op tabs, so you don't exactly know where I am or where I may have been. But as you can see, you'll see a from address, a from port, a to address and a to port with a protocol, and then packet and byte counts.



Implementing Network Security Monitoring with Open Source Tools

Statistical Data Generation

[redacted]

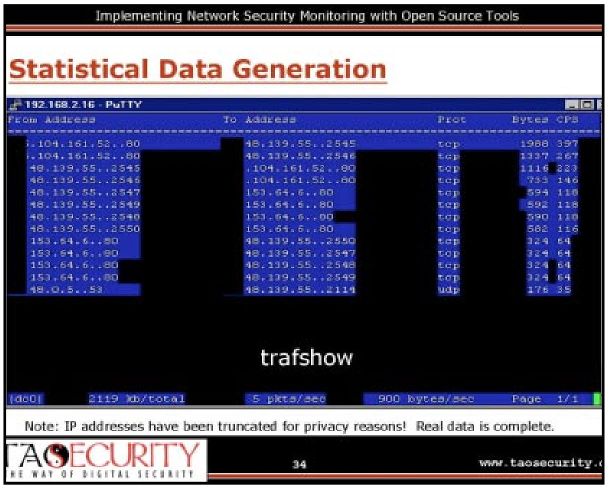
```
(dc0) trafd                              at Apr 15 08:59:51 - May  6 20:02:47
Summary: 1875891193 data bytes, 2010407821 all bytes, 4819 records
    From              Port          To               Port   Proto     Data        All
  208.157.47          554         .48.139.55         client  tcp   235269109   242939517
  12.168.18           80          .48.139.55         client  tcp    62849226    64997362
  192.0.5             443         .48.139.55         client  tcp    55806902    59027034
  .127.135.37         80          .48.139.55         client  tcp    43717973    45739981
  198.105.253         80          .48.139.55         client  tcp    43939917    45456229
  .197.170.162        80          .48.139.55         client  tcp    25666306    26371026
  8.50.29             80          .48.139.55         client  tcp    24414975    25223183
  93.38.72            80          .48.139.55         client  tcp    24014665    24805001
  158.176.214         80          .48.139.55         client  tcp    20490595    21143115
  .204.249.30         80          .48.139.55         client  tcp    20240746    20916154
  .239.159.88         80          .48.139.55         client  tcp    20154200    20838728
  22.129.85           80          .48.139.55         client  tcp    19756704    20393416
  26.73.180           80          .48.139.55         client  tcp    18647610    19264786
  153.64.7            80          .48.139.55         client  tcp    18323696    19185480
  48.139.55           client      .192.0.5           443     tcp    15206119    18107083
  .176.197.109        80          .48.139.55         client  tcp    17616633    18105513
  74.96.1             67          5.255.255.255      client  udp    16311959    17735675
  .127.92.101         80          .48.139.55         client  tcp    15495771    16018707
  .239.159.87         80          .48.139.55         client  tcp    14734086    15414830
```

Note: IP addresses have been truncated for privacy reasons! Real data is complete.

TAOSECURITY THE WAY OF DIGITAL SECURITY 33 www.taosecurity.c

Figure 33

Next slide, please. Here's trafshow. This is the real-time version. This is the one that will show you right now who is the most active in terms of bytes or packets per second and things like that.



Implementing Network Security Monitoring with Open Source Tools

Statistical Data Generation

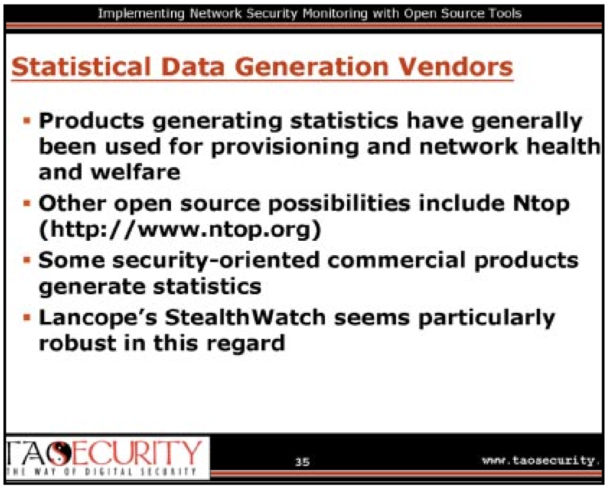
```
From Address                  To Address                      Prot        Bytes CPS
------------------------------------------------------------------------------------
[redacted]5.104.161.52..80    [redacted]48.139.55..2545       tcp          1988 397
 5.104.161.52..80              48.139.55..2546                tcp          1337 267
  48.139.55..2545              .104.161.52..80                tcp          1116 223
  48.139.55..2546              .104.161.52..80                tcp           733 146
  48.139.55..2547              153.64.6..80                   tcp           594 118
  48.139.55..2549              153.64.6..80                   tcp           592 118
  48.139.55..2548              153.64.6..80                   tcp           590 118
  48.139.55..2550              153.64.6..80                   tcp           582 116
  153.64.6..80                 48.139.55..2550                tcp           324 64
  153.64.6..80                 48.139.55..2547                tcp           324 64
  153.64.6..80                 48.139.55..2548                tcp           324 64
  153.64.6..80                 48.139.55..2549                tcp           324 64
  48.0.5..53                   48.139.55..2114                udp           176 35

                              trafshow

(dc0)     2119 Kb/total      5 pkts/sec      900 bytes/sec      Page  1/1
```

Note: IP addresses have been truncated for privacy reasons! Real data is complete.

TAOSECURITY THE WAY OF DIGITAL SECURITY 34 www.taosecurity.c

Figure 34

Could I have the next slide, please? So in terms of commercial products, I really don't know of a whole lot that do this type of data. Most of the commercial products out there are for the health and welfare of the network. Of the open source products besides trafd and trafshow, you may want to take a look at ntop. You can download it at ntop.org. It is a statistics-oriented program, but it's not exactly for security purposes; however, take a look at it. Lancope's StealthWatch is another product that is starting to show some promise in the statistics area.



Implementing Network Security Monitoring with Open Source Tools

Statistical Data Generation Vendors

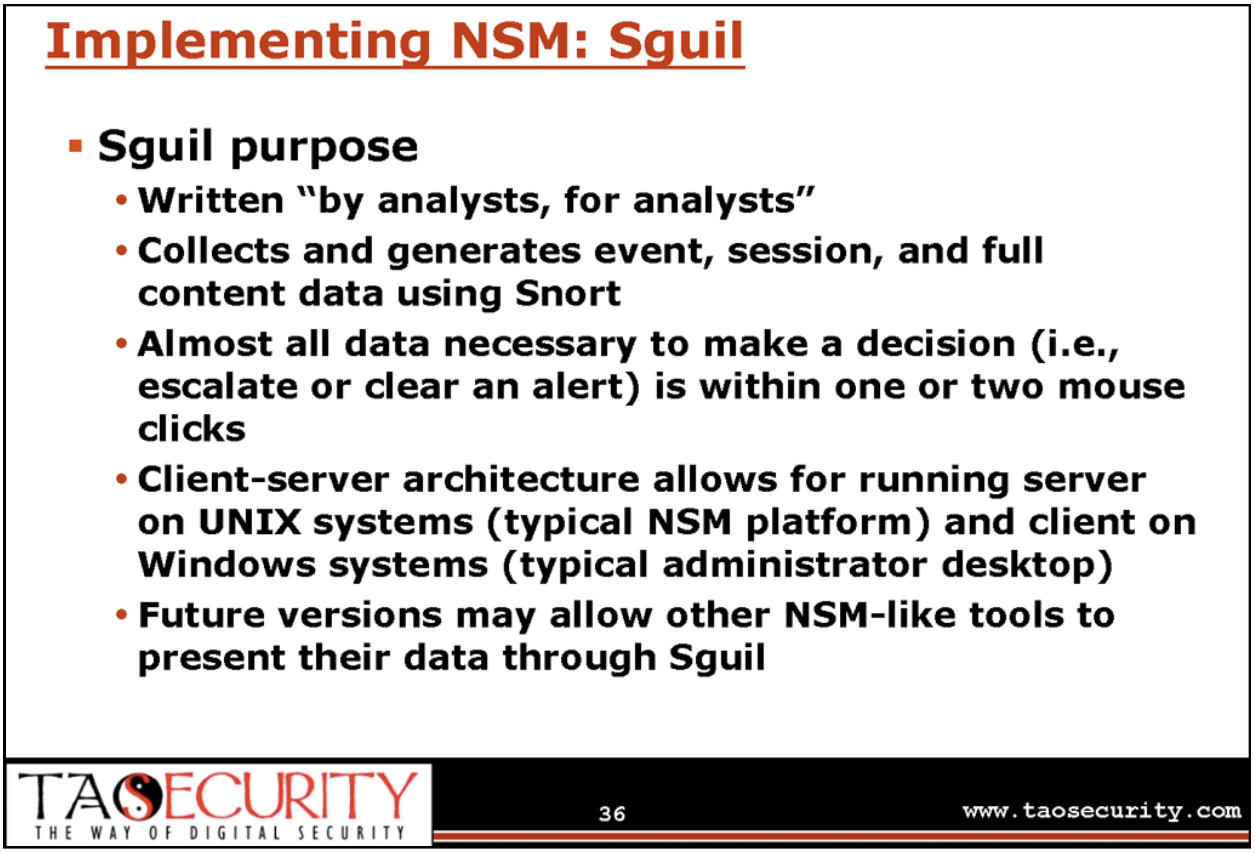
- Products generating statistics have generally been used for provisioning and network health and welfare
- Other open source possibilities include Ntop (http://www.ntop.org)
- Some security-oriented commercial products generate statistics
- Lancope's StealthWatch seems particularly robust in this regard

TAOSECURITY THE WAY OF DIGITAL SECURITY 35 www.taosecurity.

Figure 35

Could I have the next slide, please? The final tool we'll talk about is Sguil. Sguil was written by analysts, for analysts. The lead developer is a friend of mine named Bamm Visscher. We worked together at Ball Aerospace. The idea behind Sguil is to try to give you all the data you need to make a decision when you're doing network security monitoring on a single screen. And if it's not there on your screen, you only need one or two mouse clicks to get that data. It's a client server architecture, meaning you typically run the server on a Unix system. You can run the client, which is the interface to that NSM data, on a Unix box or even on a Windows box. My role with the Sguil product is writing all the documentation, which is fine, obviously. But I have written documentation on how you can access Sguil data running on a Unix server from your favorite Windows box, which is the way I do it. Bamm has also set up a demo server to which you can connect. You simply follow the instructions for setting up the client and you can take a look at the demo server.

Could I have the next slide, please? We consider Sguil to be very beta. It's only in version 0.2.5 right now. But there are complete instructions to start from scratch on the hard disk install. Build a Redhead 7.3 following the directions and you will finish with a complete Sguil install. There is work being done right now, though, to ease Sguil inflation and have it run on other platforms.

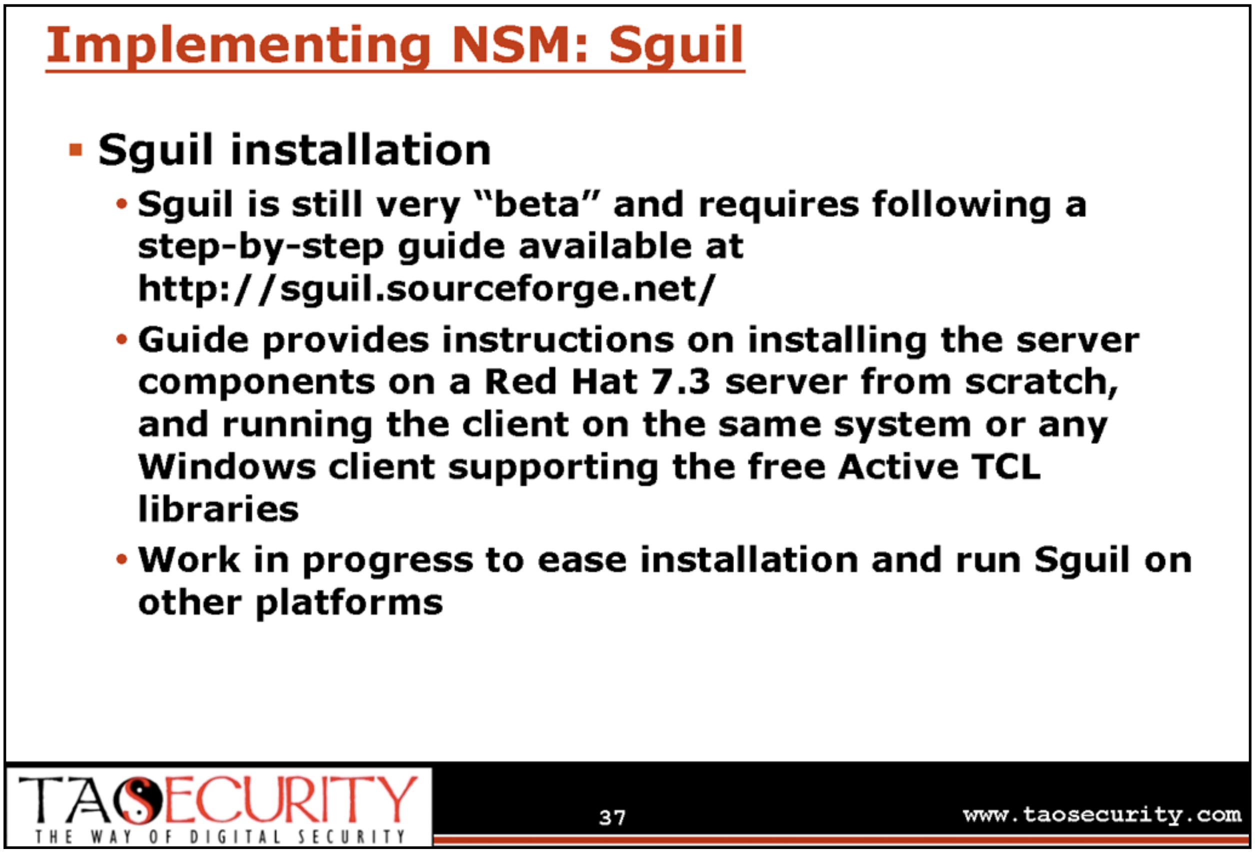


Implementing NSM: Sguil

- Sguil purpose
  - Written "by analysts, for analysts"
  - Collects and generates event, session, and full content data using Snort
  - Almost all data necessary to make a decision (i.e., escalate or clear an alert) is within one or two mouse clicks
  - Client-server architecture allows for running server on UNIX systems (typical NSM platform) and client on Windows systems (typical administrator desktop)
  - Future versions may allow other NSM-like tools to present their data through Sguil

TAOSECURITY THE WAY OF DIGITAL SECURITY 36 www.taosecurity.com

Figure 36



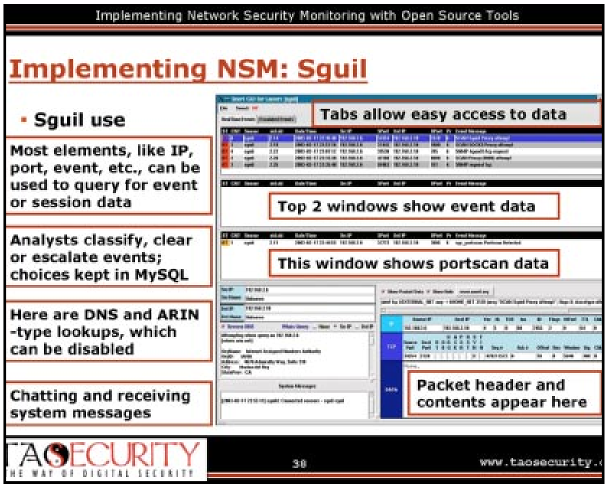
Implementing NSM: Sguil

- Sguil installation
  - Sguil is still very "beta" and requires following a step-by-step guide available at http://sguil.sourceforge.net/
  - Guide provides instructions on installing the server components on a Red Hat 7.3 server from scratch, and running the client on the same system or any Windows client supporting the free Active TCL libraries
  - Work in progress to ease installation and run Sguil on other platforms

TAOSECURITY THE WAY OF DIGITAL SECURITY 37 www.taosecurity.com

Figure 37

Could I have the next slide, please? This is just a screen capture showing Sguil. The nice thing about it is that you've got different types of data all in the same place. Let's say you've got events coming from Snort. You've got those in a couple of panes, either simply alerts or port scans. You've got full content data in the form of packets that causes the alert to fire. You've got that in the lower right-hand side. If you want session data, you can query on any of the fields — IP address, port, and so forth — and pop up session data in another tap on the Sguil interface. This session data is not collected by Argus. We use the keep stat feature of the stream port-to-processor in Snort to collect that data, but there's no reason why we couldn't modify it to collect Argus data as well. We've also got a chat interface, so all the analysts who are using Sguil at the same time can talk to each other. And, you can use Sguil to classify events by type. Is it a reconnaissance event? Is it an intrusion attempt? Is it denial of service? Is it a virus? All of this gets marked into a database they've built along with Sguil and you can query it to your heart's content.


Implementing Network Security Monitoring with Open Source Tools

Implementing NSM: Sguil

- Sguil use

Most elements, like IP, port, event, etc., can be used to query for event or session data

Analysts classify, clear or escalate events; choices kept in MySQL

Here are DNS and ARIN -type lookups, which can be disabled

Chatting and receiving system messages

Tabs allow easy access to data

Top 2 windows show event data

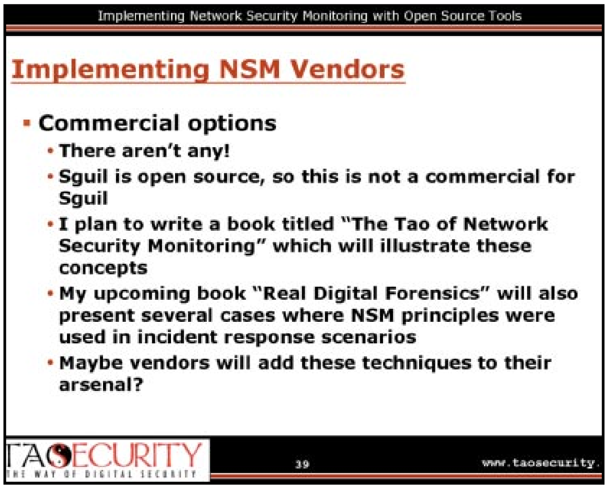
This window shows portscan data

Packet header and contents appear here

38 www.taosecurity.c

Figure 38

Could I have the next slide, please? So slide 39 is about commercial options for NSM vendors — I say there aren't any. I'm not trying to sell Sguil. This isn't a commercial for it, but there aren't any vendors who are really trying to pull all this together into one package. If you're looking for books, I am working on a book called The Tao of Network Security Monitoring, which will illustrate these concepts. I'm also working on a book called Real Digital Forensics, which will be a case-based approach to forensics and will include a couple of case studies using NSM monitoring skill sets.


Implementing Network Security Monitoring with Open Source Tools

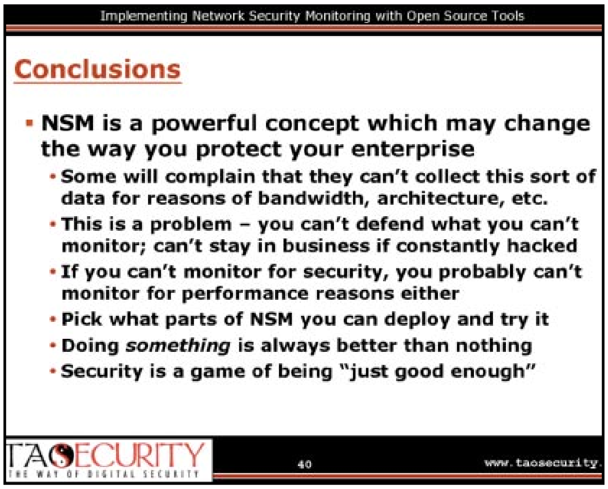
Implementing NSM Vendors

- Commercial options
  - There aren't any!
  - Sguil is open source, so this is not a commercial for Sguil
  - I plan to write a book titled "The Tao of Network Security Monitoring" which will illustrate these concepts
  - My upcoming book "Real Digital Forensics" will also present several cases where NSM principles were used in incident response scenarios
  - Maybe vendors will add these techniques to their arsenal?

39 www.taosecurity.

Figure 39

In conclusion, I'd like to think that NSM is a way that you can augment whatever work you're doing now. It doesn't replace everything that you have deployed. It doesn't rip out your current architecture. It gives you a couple other options that you can add to enhance your current collection options. The best thing to do is to pick the parts of NSM that you like and try to deploy them — and keep in mind that doing something is always better than nothing. I'd like to think of security as a game of being just good enough, and hopefully that will give you enough data to scope intermediate intrusions.


Implementing Network Security Monitoring with Open Source Tools

Conclusions

- NSM is a powerful concept which may change the way you protect your enterprise
  - Some will complain that they can't collect this sort of data for reasons of bandwidth, architecture, etc.
  - This is a problem – you can't defend what you can't monitor; can't stay in business if constantly hacked
  - If you can't monitor for security, you probably can't monitor for performance reasons either
  - Pick what parts of NSM you can deploy and try it
  - Doing *something* is always better than nothing
  - Security is a game of being "just good enough"

40 www.taosecurity.

Figure 40

And with that, I'd like to turn it back over to Crystal and take some questions.

Question & Answer Session

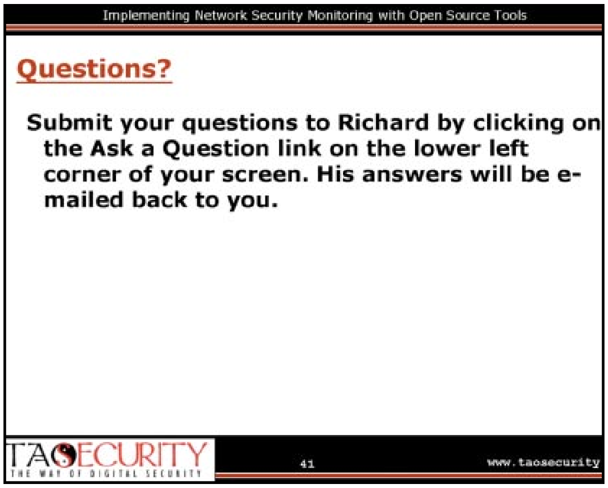
MODERATOR: Great. Thank you, Richard. You can submit your questions at any time by clicking on the "Ask a Question" link on the lower left corner of the screen.

That was a great presentation. We went a little long though, so we only have time for one or two questions.

So, I'd like to ask you, what do you think about intrusion prevention systems?

BEJTLICK: Say, you're an intrusion detection vendor and you go to a client's site to talk to the client about your product. Of course, after hearing about all your wonderful detection capabilities, the client will say, "that's nice." If you can see it, I want to stop it. Well, that's a pretty rational explanation. I mean, it's nice to know that things are happening. But it's always better to prevent them. So I think we have vendors who saw if they simply changed the middle word in IDS from detection to prevention, they could satisfy more customers and be able to say, "Yeah, we can prevent it, we can see it." The problem is in terms of simply a security model. There have always been intrusion prevention systems. There have always had firewalls. Anything that provides access control is an intrusion prevention system. The problem with IPSes, as I see it, is they have all the same faults of an intrusion detection system -- meaning if they can't see it, they can't stop it, and they have more weaknesses in terms of if they stop the wrong thing. Let's say they have a false positive and they fire on something they shouldn't have. Then, not only have they denied traffic — not only do they have that false positive — they now have denied traffic.

Now, I'm not going to totally shut them off, because right now we need this -- but people don't really do it. All ports will have to defend themselves. There just really is not a boundary anymore. We're seeing with road warriors bringing their laptops into work. You know, they're at home, they connect to the Internet, they get infected by a virus and they VPN into their organization and they start spreading a virus. So every machine will have to defend itself. So to the extent that there are products — things like systrace or other products that defend the host knowing that the host can be compromised — if they're intrusion prevention systems, then I'm all for them.



Implementing Network Security Monitoring with Open Source Tools

Questions?

Submit your questions to Richard by clicking on the Ask a Question link on the lower left corner of your screen. His answers will be e-mailed back to you.

TAOSECURITY THE WAY OF DIGITAL SECURITY 41 www.taosecurity

Figure 41

MODERATOR: So is IDS dead?

BEJTLICK: No, IDS is not dead. I'm referring to the Gartner article. There's no way that IDS is dead. The only reason that people are having trouble with IDS is that they are, in many cases, difficult to set up — and you don't get much value from them if they're not configured properly and if nobody looks at them. I completely share people's frustration with IDS. It's very frustrating to see an event fire and have no idea what it is, what caused it or what to do next. That's the whole reason I came up with this NSM presentation and why I'm writing a book about it. The idea behind NSM is to ask, what do I do next? Where did this event come from? What else is related to it? What else has been happening? What sort of supporting evidence do I have to make a decision and to scope and remediate this incident? So to the extent that we can supplement IDS with these network security monitoring techniques, I think we'll all be happier.

MODERATOR: Great. Thank you for your insight, Richard. Do you have any final comments for our listeners?

BEJTLICK: No. Thank you very much for attending.

MODERATOR: Thank you. This concludes today's webcast with Richard Bejtlick. If you have any comments on this webcast or suggestions for future webcasts, please e-mail us at: webcast@searchsecurity.com.

Before we go, I'd like to encourage you to consider attending Information Security Magazine's Security Decisions conference hosted by SearchSecurity.com in Chicago October 15th through the 17th. We have a great lineup of speakers who will provide you with expert insights, and attendance is free for those who qualify. Stop by www.securityconf.com to learn more. I hope to see you there. Thanks again to today's sponsor, Sprint, and to our guest speaker, Richard Bejtlick — and thank you for joining us. Have a great rest of the day.

Implementing Network Security Monitoring with Open Source Tools

Thank you

Thank you for participating in this SearchSecurity.com on-demand webcast. If you have comments on this webcast or suggestions for future webcast topics, please send an e-mail to webcast@searchSecurity.com.

Figure 42

About TechTarget

We deliver the information IT pros need to be successful.

TechTarget publishes targeted media that address your need for information and resources. Our network of industry-specific Web sites gives enterprise IT professionals access to experts and peers, original content and links to relevant information from across the Internet. Our conferences give you access to vendor-neutral, expert commentary and advice on the issues and challenges you face daily. Practical technical advice and expert insights are distributed via more than 100 specialized e-mail newsletters, and our Webcasts allow IT pros to ask questions of technical experts in real time.

What makes us unique

TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to IT professionals. We leverage the immediacy of the Web, the networking and face-to-face opportunities of conferences, the expert interaction of Webcasts and Web radio, the laser-targeting of e-mail newsletters and the richness and depth of our print media to create compelling and actionable information for enterprise IT professionals. For more information, visit www.techtarget.com.

About SearchSecurity.com

Information Security magazine is now part of TechTarget, bringing together the media leaders in the information security space. SearchSecurity.com and Information Security Magazine provide IT security professionals with the information they need to keep their corporate data and assets secure. SearchSecurity.com and InfoSec Magazine are the essential resources for IT security professionals.