Argus – Auditing network activity

By Russ McRee

Prerequisites

*nix OS with bison, flex, and libpcap
Argus-clients-3.0 to utilize racluster
Graphviz to work with neato for AfterGlow output
RRDtool (RRDs.pm) for ragraph

Similar or inclusive projects

HeX liveCD includes Argus[1]
Anemone[2]

Argus[3] is the network Audit Record Generation and Utilization System, a Real Time Flow Monitor that is designed to perform comprehensive data network traffic auditing. The brainchild of Carter Bullard of QoSient, "The Argus Open Project is focused on developing network activity audit strategies that can do real work for the network architect, administrator and network user." As longtime advocate for good network security monitoring (NSM) tactics, I was first exposed to Argus via Bejtlich's The Tao of Network Security Monitoring. Consider this essential reading, if you have not already read it. Where Argus shines for yours truly is, of course, security assurance. Bejtlich considers Argus "the single most important tool in the emergency NSM arsenal." Forced to choose one application in an incident response scenario, it would be Argus.[4]

When reading up on Argus, refer directly to its website.[5] Web searches for Argus will also yield another project unfortunately of the same name that has nothing to do with this fine NSM offering.

Project details

First iterations of Argus are traced back to 1990 when Bullard put it to use for investigative purposes while a grad student at Georgia Tech. A few years later Cisco's NetFlow debuted, but with one notable difference. Where NetFlow is unidirectional, Argus is a bi-directional flow modeler, matching network responses to any network traffic that is sent. In adhering to the IETF's Framework for IP Performance Metrics, Argus matches multiple identifier/descriptors in various flow models in Layers 2 through 5 of the OSI. Updated regularly since its inception, and with a strong, supportive community, Argus is nearing a final 3.0 release. I tested a 3.0 release candidate in order to take advantage of new client features described in the *Clients* section. This release, slated for availability by year's end, will offer IPv6 capabilities, arbitrary encapsulation parsing (VLAN, MPLS, GRE, IPnIP) and can be quickly adapted to new protocols, sometimes providing basic metrics without extension.

According to the NSMWiki, you will find universities using Argus to record both internal traffic flows, as well as flows outside the DMZ to detect infected or compromised machines, in real time. Argus is in use at U.S. government facilities to provide more extensive network forensics, and is the focus of research in control-plane network non-repudiation. The new release provides the parts to build a distributed network activity audit system for the complete enterprise; Argus as a data source installed in the end-system and as a network-based flow monitor, and the client program *radium*, which is a flow data collection and distribution system. Argus client programs can also read and convert NetFlow data, so you do not have to ignore that as a flow data source. Finally, network research labs have used Argus to test network performance of unique protocols, such as Infiniband over IPv6.[6]

Bullard describes Argus on his (QoSient) website as well suited to security assurance as it "enables the establishment of a comprehensive audit trail of all network activity, either for a single network element or for an entire network segment." Opportunities abound, including:

- Non-repudiation
- Incident response
- Policy enforcement validation
- Protocol validation
- Network behavioral baselining
- Intrusion detection
- Discovery detection
- Network asset inventory

If we agree that the definition of incident response is "the practice of detecting a problem, determining its cause, minimizing the damage it causes, resolving the problem, and documenting each step of the response for future reference,"[7] then incorporating Argus is the embodiment of this endeavor.

1 http://www.rawpacket.org/projects/hex-livecd.

2 https://research.microsoft.com/projects/anemone/

3 http://qosient.com/argus/index.htm.

4 Richard Bejtlich, *The Tao of Network Security Monitoring,* Addison-Wesley, 2005, Pg. 236.

5 http://qosient.com/argus/index.htm

6 Argus-NSMWiki, http://www.vorant.com/nsmwiki/index.php?title=Argus

7 https://wiki.internet2.edu/confluence/display/secguide/Glossary.

Installation

Argus is simple to compile and install. You can grab source and `./configure, make && make install`, assuming your dependencies are met (bison, flex, libpcap), and has been ported to most platforms, including Cygwin, and OpenWrt. Binary packages are also available.

Server

My use of Argus is most often on a Linux server listening to a SPAN port, but any "network tapping strategy that captures all the packets destined to and from the target(s)"[8] will suffice.

Keep in mind though that a SPAN port has limitations and that you are well advised to utilize a well placed tap. Lots of conventional wisdom is available to you if you search span versus tap.

Keep in mind, Argus can be deployed directly on a server of interest to measure performance itself.

I'll show a couple of specific use scenarios, but I'd also refer you to an excellent resource that provides precise details on Argus use. *Structured Traffic Analysis*, by Richard Bejtlich, can be found in *(IN)SECURE Magazine*, Issue 4.[9] Richard covers an entire process for network incident response (NIR) that includes other useful tools in addition to Argus, but provides an ideal play-by-play of Argus use as well.

There is a configuration file (most often `/etc/argus.conf`) wherein you can daemonize it, set a PID, manage instances, and set the listening port, IP, and interface. All settings have command line equivalents, conveniently explained in the configuration sample, as well as the man pages, and `argus –h` will always come through for you.

For this demonstration, I have set Argus to run simply. I am not using it to measure performance, so I will not pass parameters like `–R` for response time data; but remember Argus excels in this capacity. I just want it to listen on *eth1* and write to *toolsmith.out* so I've passed `argus –I eth1 –w toolsmith.out` at the prompt.

Security considerations

If you set Argus up for remote connectivity, you have some security considerations to attend to. Access control can be managed by tcp_wrappers where you can specify what hosts can access Argus, or you can incorporate the Simple Authentication and Security Layer (SASL). SASL provides strong authentication and confidentiality protection for Argus data on the wire, deemed "very important stuff when accessing remote real-time Argus data."[10]

Clients

With my freshly acquired output file I have a number of client options to run the data through. Once you have downloaded the client tar and compiled and installed, you should have tools like *ra, ragrep, racount, rasort*, and *ragraph* at your disposal. As an example I ran `ra –n –r toolsmith.out – udp` to see what kind of UDP was being generated and spotted a destination port of 9996. This can be interesting from a malware perspective but also from a NetFlow perspective. A closer look yielded:

```
# ra -n -r toolsmith.out - udp dst port 9996
07 Oct 07 20:46:10   F  udp  172.16.130.11.51310  ->  172.16.34.22.9996  1  0  58     0  INT
07 Oct 07 20:46:23   F  udp  172.16.66.253.49741  ->  172.16.34.22.9996  2  0  1540   0  INT
07 Oct 07 20:46:28   F  udp  172.16.30.254.56003  ->  172.16.34.22.9996  1  0  1490   0  INT
07 Oct 07 20:46:13      udp  172.16.6.11.52770    ->  172.16.34.22.9996  19 0  28614  0  INT
07 Oct 07 20:46:14   F  udp  172.16.30.253.53223  ->  172.16.34.22.9996  6  0  7500   0  INT
07 Oct 07 20:46:19      udp  172.16.6.12.50443    ->  172.16.34.22.9996  11 0  16566  0  INT
07 Oct 07 20:46:54   F  udp  172.16.30.254.56003  ->  172.16.34.22.9996  2  0  2980   0  INT
07 Oct 07 20:46:26   F  udp  172.16.130.10.57488  ->  172.16.34.22.9996  17 0  23890  0  INT
07 Oct 07 20:46:56   F  udp  172.16.130.11.51310  ->  172.16.34.22.9996  6  0  290    0  INT
```

Hmm…multiple hosts sending to a single IP and dport 9996. A quick nmap scan discovered port 8080 open on this host which, when browsed, produced a NetFlow Analyzer web portal. The irony of this discovery via Argus should not be lost on NSM practitioners.

For an interesting read on tracking specific traffic, like bot-infected hosts, check out the Argus-PracticalBotNetDetection.pdf.[11]

For you OCD types (yeah, me too) who need their lists presented in an orderly fashion, *ra* is extended to meet such needs with *rasort*. A snap shot shows Kerberos and pop3 all sorted nicely:

```
07 Oct 07 20:46:43      udp  172.16.37.30.15158  <-> 172.16.30.12.kerbe  1  1  332  1415  CON
07 Oct 07 20:47:03   d  tcp  172.16.37.30.15198  <-  172.16.30.12.kerbe  0  2  0    124   ACC
07 Oct 07 20:47:07      tcp  172.16.90.120.1640  ?>  172.16.30.11.kerbe  2  3  242  1494  RST
07 Oct 07 20:47:07      udp  172.16.34.15.3391   ->  172.16.30.13.kerbe  1  0  1420 0     INT
07 Oct 07 20:47:11      udp  172.16.34.15.3416   ->  172.16.30.13.kerbe  1  0  1420 0     INT
07 Oct 07 20:46:51      udp  172.16.91.48.3312   <-> 172.16.30.11.kerbe  1  1  1396 1396  CON
07 Oct 07 20:46:20   s  tcp  172.16.40.60.2980   ->  172.16.31.143.pop3  2  2  124  108   FIN
07 Oct 07 20:46:20      tcp  172.16.40.60.2982   ?>  172.16.31.143.pop3  3  1  194  54    FIN
07 Oct 07 20:46:21      tcp  172.16.40.60.2991   ?>  172.16.31.143.pop3  4  0  222  0     FIN
07 Oct 07 20:46:21      tcp  172.16.40.60.3001   ?>  172.16.31.143.pop3  2  1  134  88    FIN
07 Oct 07 20:46:21      tcp  172.16.40.60.3015   ?>  172.16.31.143.pop3  2  0  135  0     FIN
07 Oct 07 20:46:21      tcp  172.16.40.60.3020   <?> 172.16.31.143.pop3  1  0  54   0     TIM
```

If you make use of RRDtool, *ragraph* will offer you a graphical representation of Argus output as well. Issuing `ragraph bytes –M 1s –fill –stack –r toolsmith.out – udp and port 9996` will produce Figure 1.

Visualization opportunities

Taking the visual focus to the next level, there are certain visualization projects that can work with Argus data as well. If

8 http://qosient.com/argus/how-to.htm#8

9 www.net-security.org/dl/insecure/INSECURE-Mag-4.pdf.

10 http://qosient.com/argus/faq.htm#11.1

11 http://www.rawpacket.org/anonymous/papers/Argus-PracticalBotNetDetection.pdf.

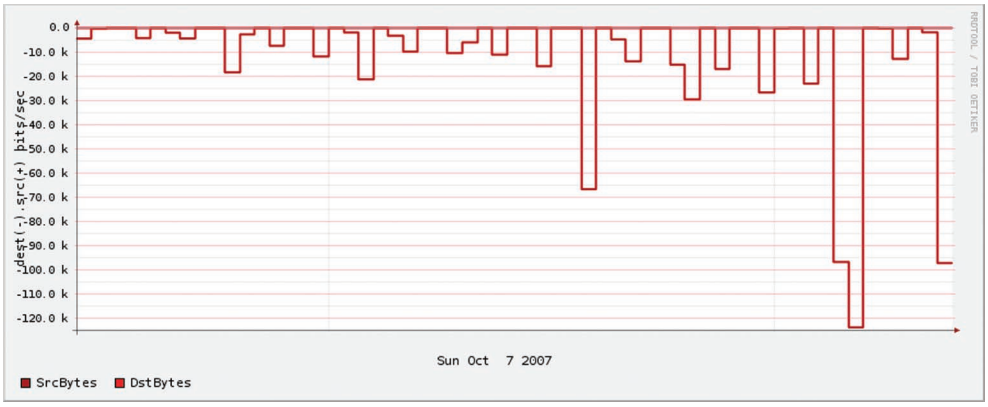

Figure 1 – Graphical representation of Argus data

security visualization is of interest to you, check out secviz.org. There is also a new book out by Greg Conti called *Security Data Visualization* and part of a nice review of it at the TaoSecurity blog.

Argus output can be rendered by certain visualization projects, in particular AfterGlow,[12] recently updated to version 1.5.9. Argus 3.0 includes a number of new client features that aid in the visualization process, including direct output to CSV (needed for AfterGlow), *ranonymize*, which will change your IPs in the Argus output to create privacy, and *racluster*, an aggregator, both of which can be utilized for AfterGlow visualizations. A 60 second capture from a very busy network, randomized for privacy, can be rendered as follows, resulting in Figure 2:

```
ranonymize -r toolsmith.out -w - | racluster
-r - -m saddr daddr dport -c, -s saddr daddr
- 'tcp and dst port 25' | /opt/afterglow/
src/perl/graph/afterglow.pl -t -e 2 -c /opt/
afterglow/src/perl/parsers/color.properties |
neato -Tgif -o tcp25argus.gif
```

SecViz.org provided an excellent starting point for me, combining use of Argus, AfterGlow, and Neato.[13]

Benefits and drawbacks

The only time I could imagine a drawback when using Argus might be in a scenario where a heretofore unmonitored user LAN is graced with a first look from an NSM implementation including Argus. The resulting horror of seeing what previously unmonitored users are up to would count as a drawback, given the probable heart attack for the analyst.

Conversely, the same information would be considered highly beneficial as it would aid the enterprise in question in the process of improving its security posture. You cannot fix what you cannot see. Risk reduced is confidence gained.

In conclusion

It has been said the Argus is easy to use but hard to master, and you may find that an honest assessment. But the references included at the end of this discussion will quickly lead you to further discovery. Regardless, consider Argus essential as part of your situational awareness arsenal, in both performance and security capacities.

Carter would love feedback from the Argus community, from longtime users to those new to Argus, as well as those who would like to provide testing for the pending 3.0 release.

Enjoy this tool – it is extremely useful. And for NSM practitioners, it is nirvana. Cheers…until next month.

Acknowledgments

—Carter Bullard, for his years of work on Argus, as well as his feedback and insight for this article.

—Richard Bejtlich, for paving the NSM way, www.taosecurity.com.

—David Bianco, for the NSMWiki.

—CS Lee (geek00l), geek00l.blogspot.com, for his endless dedication to the NSM cause.

About the Author

Russ McRee, GCIH, CISSP, is a security analyst working for Expedia. He is a member of numerous security organizations and maintains holisticinfosec.org. Contact him at russ@holisticinfosec.org.

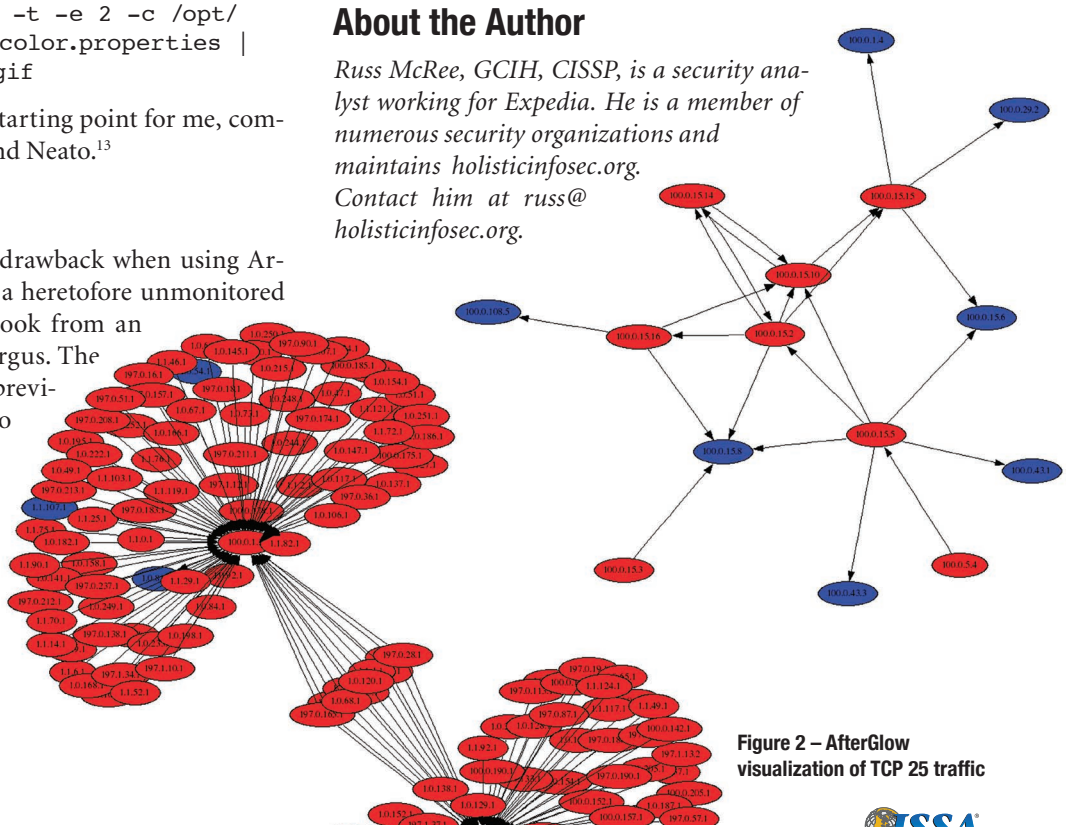

Figure 2 – AfterGlow visualization of TCP 25 traffic

12 http://afterglow.sourceforge.net

13 http://secviz.org/?q=node/74.

