NetGrok and AfterGlow:

Visualizing the Zeus attack against government and military

By Russ McRee – ISSA member, Puget Sound (Seattle), USA Chapter

Prerequisites

Java Runtime Environment for NetGrok

Linux distribution for AfterGlow (see DAVIX)

When last we discussed security visualization for *toolsmith* in June 2008.[1] DAVIX[2] (Data Analysis and Visualization Linux) was just gearing up for its initial release at Black Hat/DEFCON and the development of NetGrok was just wrapping up in Ben Shneiderman's Information Visualization class at University of Maryland. Today, NetGrok is expected to be included in the next release of DAVIX, and both NetGrok and DAVIX have been well received by the security data visualization community.

Motivation for this month's topic comes from the fact that I'll be presenting *Visualizing IDS Output: Tools and Methodology* for RSA 2010 attendees on March 5, 2010 at 1010 hours.

Following is the session abstract:

"The flood of raw data generated by intrusion detection systems (IDS) is often overwhelming for security specialists, and telltale signs of intrusion are sometimes overlooked in all the noise. Security visualization tools provide an easy, intuitive means for sorting through the dizzying data and spotting patterns that might indicate intrusion…the presentation will focus on specific tools and methodology to aid you in establishing security data visualization practices in your environment."

I'll be discussing both tools discussed in this month's column, as well as Maltego (discussed in December 2009 as the 2009 Toolsmith Tool of the Year[3]), including live demonstrations. I'll accentuate this theme as the crux of our *toolsmith* discussion this month while discussing NetGrok and AfterGlow and additionally introduce timely sample analysis of the targeted Zeus bot attacks in early February against U.S. government institutions.

From Brian Kreb's article[4] on the "2020 Project" attack:

> *"The messages are spoofed so that they appear to have been sent by the National Intelligence Council (address used was nic@nsa.gov), which serves as the center for midterm and long-range strategic thinking for the U.S. intelligence community and reports to the office of the Director of National Intelligence.*
>
> *The e-mails urge recipients to download a copy of a report named "2020 Project." Another variant is spoofed to make it look like the email came from admin@intelink.gov. The true sender, as pulled from information in the email header, was nobody@sh16.ruskyhost.ru."*

Any victim engaging the link sent in the attacker's email was treated to 2020.exe.

We'll first conduct static Snort analysis of a PCAP (zeus.pcap) taken while monitoring an infected a Windows XP victim virtual machine with 2020.exe; I'll then utilize the same PCAP to exemplify the benefits of NetGrok and AfterGlow.

Zeus/Zbot binary

2020.exe (MD5: 3cfc97f88e7b24d3ceecd4ba7054e138), as distributed in the above mentioned targeted email flood, is well identified as this is being written.[5] ThreatExpert's blog includes a detailed analysis of Zeus/Zbot; I'll refer you there[6] rather than repeat content. The takeaway: Zeus/Zbot is an annoying, persistent, thieving, harmful little bugger and is best avoided at all costs. ;-)

2020 sample PCAP analysis via Snort

There should be no surprises here. Ensure that you've added the latest Emerging Threats signatures to your static Snort installation rules directory and enabled them in `snort.conf`. Be cautious using them for real-time analysis; left untuned `emerging-all.rules` can cause a lot of packet drop on a busy sensor. For static analysis though, enable everything you've got and see what pops. ;-)

Run `sudo snort -c /etc/snort/snort.conf -r zeus.pcap -l output/zeus2020` after creating the *output/zeus2020* directory. I'll send you this PCAP on request if you'd like to experiment with it. The resulting alerts are seen in Figure 1.

You'll note that our unwelcome visitor connects to 115.100.250.105[7] in China (wow, shocking…really?), which in turn resolves as *updatekernel.com*. I've got a kernel for you to update right here. As I said, no real surprises here, just normal targeted malware attacks emanating from China. Old

[1] https://holisticinfosec.org/toolsmith/docs/june2008.pdf.

[2] http://www.secviz.org/node/89.

[3] http://holisticinfosec.org/toolsmith/docs/december2009.html.

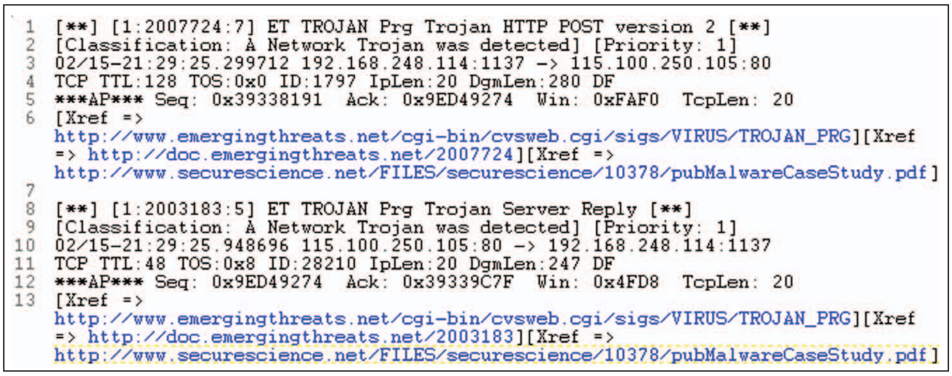
[4] http://www.krebsonsecurity.com/2010/02/zeus-attack-spoofs-nsa-targets-gov-and-mil.

[5] http://www.virustotal.com/analisis/82d10922cc1365a79b43a16502211ae610f56b01cd36a18db67d8a0c81c434c4-1266285240.

[6] http://blog.threatexpert.com/2009/09/time-to-revisit-zeus-almighty.html.

[7] https://zeustracker.abuse.ch/monitor.php?host=updatekernel.com.

Figure 1 – Zeus/Zbot as identified by Snort & Emerging Threats rules.



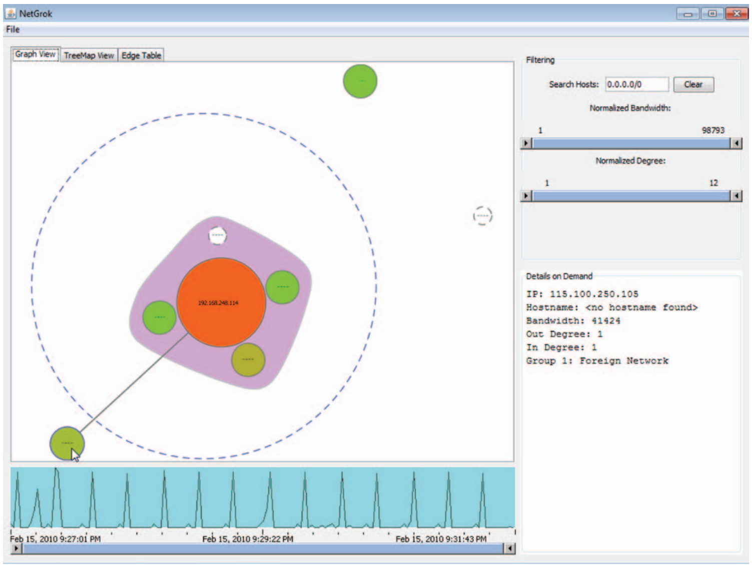
```
[**] [1:2007724:7] ET TROJAN Prg Trojan HTTP POST version 2 [**]
[Classification: A Network Trojan was detected] [Priority: 1]
02/15-21:29:25.299712 192.168.248.114:1137 -> 115.100.250.105:80
TCP TTL:128 TOS:0x0 ID:1797 IpLen:20 DgmLen:280 DF
***AP*** Seq: 0x39338191  Ack: 0x9ED49274  Win: 0xFAF0  TcpLen: 20
[Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_PRG][Xref
=> http://doc.emergingthreats.net/2007724][Xref =>
http://www.securescience.net/FILES/securescience/10378/pubMalwareCaseStudy.pdf]

[**] [1:2003183:5] ET TROJAN Prg Trojan Server Reply [**]
[Classification: A Network Trojan was detected] [Priority: 1]
02/15-21:29:25.948696 115.100.250.105:80 -> 192.168.248.114:1137
TCP TTL:48 TOS:0x8 ID:28210 IpLen:20 DgmLen:247 DF
***AP*** Seq: 0x9ED49274  Ack: 0x39339C7F  Win: 0x4FD8  TcpLen: 20
[Xref =>
http://www.emergingthreats.net/cgi-bin/cvsweb.cgi/sigs/VIRUS/TROJAN_PRG][Xref
=> http://doc.emergingthreats.net/2003183][Xref =>
http://www.securescience.net/FILES/securescience/10378/pubMalwareCaseStudy.pdf]
```

news for most of us, but if we apply visualization techniques we can utilize this malicious traffic to enhance perspective.

2020 sample PCAP analysis via NetGrok

NetGrok is a standalone Java application that works on any Java-supported operating system. Download the installation package and utilize the .bat startup file on Windows or issue `java –jar netgrok20080928.jar` on a *nix system.

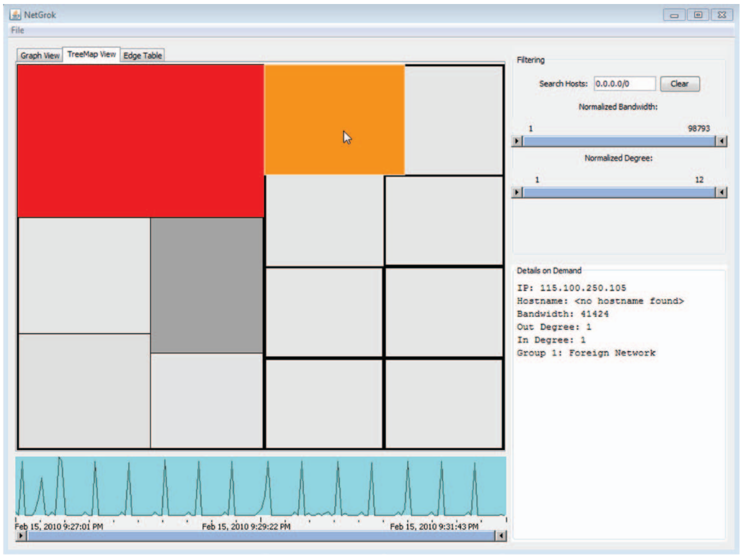
Once NetGrok is running you will find a simple and intuitive user interface. To analyze zeus.pcap I simply clicked *File => Open Pcap File* (Ctrl+P). While the host count found in `zeus.pcap` is not at all large (my test environment is small; you'll note only a few connections) this PCAP does exemplify how quickly a top talker can be identified. There are three views offered by NetGrok, the most important are Graph View and TreeMap View. Both offer features that cause top talkers to jump off the screen at you. You'll note in Figure 2 a few key items. The dashed circle indicates the boundary between internal and external hosts. The color indicates the extent to which a given host is communicating. The large central red circle indicates the infected host. As the color darkens from green to red, the color change is related to the amount of traffic to and from the host in question. The purple blob indicates a private network. You'll also note that by hovering over a host, its Details on Demand are populated in the lower right-hand pane. This data will update as you move your pointer amongst hosts. Double-clicking a host will also cause the view to zoom in, displaying the IP address. If you right-click on the same zoomed host you can do a DNS lookup on the IP. At the bottom of the NetGrok UI you'll see what appears as an EKG, which is quite like a host heart beat monitor. In this case it's the infected host and the Zeus C&C server checking in with each other via HTTP. Imagine how useful these quick identification methods might be with a very noisy PCAP inclusive of many hosts.



Filtering
Search Hosts: 0.0.0.0/0 Clear
Normalized Bandwidth: 1 98793
Normalized Degree: 1 12

Details on Demand
IP: 115.100.250.105
Hostname: <no hostname found>
Bandwidth: 41424
Out Degree: 1
In Degree: 1
Group 1: Foreign Network

Figure 2 – Zeus/Zbot and victim communication as identified by NetGrok Graph View.



Filtering
Search Hosts: 0.0.0.0/0 Clear
Normalized Bandwidth: 1 98793
Normalized Degree: 1 12

Details on Demand
IP: 115.100.250.105
Hostname: <no hostname found>
Bandwidth: 41424
Out Degree: 1
In Degree: 1
Group 1: Foreign Network

Figure 3 – Zeus/Zbot and victim communication as identified by NetGrok TreeMap View.

The TreeMap View also allows one to cut to the quick. This view divides hosts up in classic block-form TreeMap fashion, following color gradation principles similarly to GraphView. Hosts represented in the left hand side of the TreeMap UI are private IP, local hosts; hosts on the right are external. Much as noted in Figure 2, the big red block on the left is my infected VM, and the darkening block to its immediate right is the Zeus server. Hovering over any block will populate Details on Demand as expected (figure 3).

Again, imagine how quickly you can spot the bad guy when analyzing a large, busy PCAP inclusive of many hosts.

2020 sample PCAP analysis via AfterGlow

Detailed references on how to make use of AfterGlow are available via the manual on the Sourceforge site.[8] The important highlights follow.

[8] http://afterglow.sourceforge.net/manual.html.

AfterGlow works from CSV files created from PCAPs. Other visualization tools included on DAVIX do as well, so I often do the conversion as a standalone process by piping tcpdump output to the `tcpdump2csv.pl script`. For the `zeus.pcap` file I executed the following:

```
tcpdump -vttttnnelr zeus.pcap | /usr/local/bin/
tcpdump2csv.pl "sip dip dport" > zeus.csv
```

Any number of parameters can be defined: source IP (sip), destination IP (dip), source port (sport), destination port (dport), time-to-live (TTL), etc. `Cat tcpdump2csv.pl` to determine what options suit you. I love tweaking these options; *timestamp* is a real bonus forensically.

To generate a dot graph file for Graphviz,[9] execute:

```
cat file.csv | perl afterglow.pl -c color.
properties > file.dot
```

This file can then be used with *dot* or *neato* to render a graph.

As mentioned above, I usually separate CSV creation and graph creation, but choosing to combine all efforts, one can generate a graph (gif file) from a PCAP file as follows (see Figure 4):

```
tcpdump -vttttnnelr /home/rmcree/zeus.pcap
| ./tcpdump2csv.pl "sip dip dport" | perl
afterglow.pl -c color.properties | neato -Tgif
-o zeus.gif
```

You'll note a color property file is specified in the AfterGlow call. This file is used by AfterGlow to determine the colors of the edges and nodes in the graph. Said property file can be customized to your preferences.

In Figure 4 red indicates external entities, yellow represents the local private network, dark blue denotes a TCP destination port, and light blue is a UDP destination port. Our infected host chatting with its Chinese C&C server (115.100.250.105) jumps right out at you. Full packet analysis of the PCAP indicated that my infect VM downloaded its config file, `x98x10.bin`, from updatekernel.com (115.100.250.105) via port 80.

Figure 4 will remind you how useful a good picture can be to bring you to the heart of the matter. ;-)

I recommend performing all these functions on the DAVIX virtual machine you should create from the ISO. AfterGlow scripts are inherent to the AfterGlow shell, called from the menu via `KDE Start → Visualize → AfterGlow`, and executed from */usr/local/bin*, thus eliminating the need to call Perl.

In conclusion

I hope to see you at RSA; I'll do live demonstrations of NetGrok, AfterGlow, and others for you.

Be sure to read the two "bibles" of security data visualization: Greg Conti's Security Data Visualization[10] and Raffael Marty's Applied Security Visualization.[11] Ping me via email[12] if you'd like me to send you the Zeus PCAP or any other related files.

Make good use of security data visualization to help you more quickly identify malicious traffic; as it has me, it will serve you well.

Cheers…until next month.

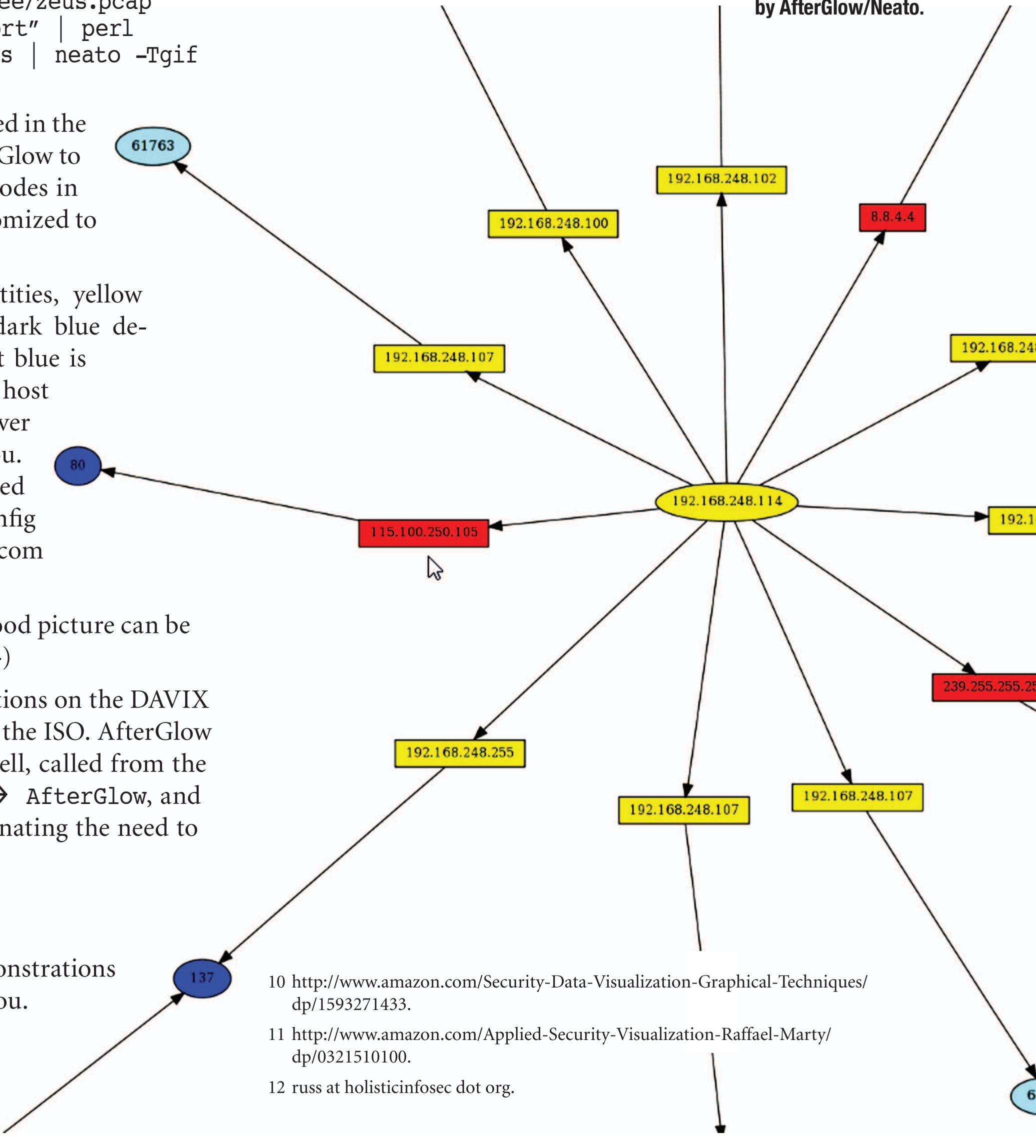
Acknowledgments

—Greg Conti, Raffael Marty, Jan Monsch

About the Author

Russ McRee, GCIH, GCFA, GPEN, CISSP, is team leader and senior security analyst for Microsoft's Online Services Security Incident Management team. As an advocate of a holistic approach to information security, Russ' website is holisticinfosec.org. Contact him at russ@holisticinfosec.org.

Figure 4 – Zeus/Zbot and victim communication as identified by AfterGlow/Neato.



[9] http://www.graphviz.org.

[10] http://www.amazon.com/Security-Data-Visualization-Graphical-Techniques/dp/1593271433.

[11] http://www.amazon.com/Applied-Security-Visualization-Raffael-Marty/dp/0321510100.

[12] russ at holisticinfosec dot org.