Managing Badware and Policy Violation with Aanval and Bleeding Edge Threat Snort Rules

By Russ McRee

Introduction

Badware, instant messaging (IM), and peer-to-peer (P2P) apps, are prominent issues facing those with the task of securing enterprise computing environments. The risks are plentiful, including potential loss of confidential data, Trojans, and rootkits. Our focus this month will detail the effort to manage those risks on your network using *Aanval,* an excellent *Snort* console from Remote Assessment, and *Bleeding Threat Snort Rules* designed to readily identify traffic of concern.

We won't cover installing Snort here, with the exception of stating that you need to be running snort-mysql because of a heavy MySQL dependency for this project. Additionally, the configuration of your network environment to support Aanval and Snort is the subject of many fine books and articles, so we won't spend time there either.

However, we will spend a great deal of time with Aanval.

According to their website, Aanval (pronounced: "anvil") is an advanced data management, correlation and analysis console designed specifically for Snort and Syslog data. Aanval is a complete web-based software solution designed to manage and correlate snort intrusion detection data and/or syslog device data. It is available in a free, downloadable, single-sensor version in addition to a commercially licensed and supported version. Aanval offers a plethora of functionality, including an extensive query capacity and attractive reporting. Our interest here is purely Snort, and specifically the use of Bleeding Edge Threats rules.

Imagine being able to report to your management team the current state of your network regarding spyware-infected hosts, and then offering an eradication and protection plan. Perhaps your organization has banned the use of IM in a policy. Or, as in my darkest nightmare, a user has gone full-goose bozo, installed the latest Google Desktop and is sharing personally identifiable information across multiple Pcs – beyond the scope of your administrative control. In an age of compliance, it is hugely beneficial to have the capacity to draw the majority of network security information from one platform. In my use of Aanval I've gained as much information about outbound traffic from spyware and IM as I do from typical inbound threats.

Imagine being able to report to your management team the current state of your network regarding spyware-infected hosts, and then offering an eradication and protection plan.

General server configuration

While options abound, we'll keep the server configuration a simple one: httpd, MySQL, and Snort are all on one server. Additional tools like Barnyard and Oinkmaster are available and offer great benefit, but for the sake of this column, we'll be keeping the focus on badware and policy violation detection rather than server configuration. But first, on to some essential administrivia.

Snort

Let's begin with Snort. Again, we'll assume you're familiar with Snort installations and may already be using a BASE console or something similar. If you're new to Snort, consider Patrick Harper's install doc.[1] Follow his how-to's standard installation right up to page 13 where he begins to describe BASE installation. Not to knock BASE, but we're going to significantly surpass its capabilities with Aanval. The most important variable in your Snort installation is this: `./configure –with-mysql make, make install`. Without the `–with-mysql` parameter this effort is for naught. Note: If you are using BASE, keep in mind that BASE uses its own DB structure that it writes to the Snort DB. It has been my experience that these tables don't play well with Aanval, thus, I suggest a fresh, previously unmodified Snort DB for this process.

1 http://internetsecurityguru.com/

Bleeding Edge Threats

Key ingredients in this mix are Bleeding Edge Threat signatures. Easily attainable from http://bleedingthreats.net, these signatures are updated on a near daily basis. These signatures are divided into individual rulesets such as malware or P2P. For this use, however, I prefer using bleeding-all.rules, which compiles them all into one set. It's easier to manage in one file as it minimizes rule set entries in your `snort.conf`. Perhaps you're an Oinkmaster user, as I am; but I update Bleeding Edge rules a bit differently. Oinkmaster requires a .tar.gz file and the bleeding-all.rules file doesn't conform. Plus, the .tar.gz rule set will break the rules out into all their categories rather than the monolithic file. I grab it manually like this:

```
cd /etc/snort/rules
wget http://www.bleedingthreats.net/bleeding-all.rules
```

There are occasions when a rule may hang Snort up due to a missing variable or a typo.

To test that all things Snort will run smoothly, enter `/usr/local/bin/snort -c /etc/snort/snort.conf -i eth1 -v`. If all is well, signatures will begin to roll by after initialization. If not you'll get an error message, perhaps like `ERROR: Undefined variable name: (/etc/snort/rules/bleeding-all.rules:2533): SSH_PORTS`. This tells you to go to line 2533 of the bleeding-all.rules file and address the issue. You could define the variable or comment out the error causing line. Repeat the process until all is working satisfactorily. After you've confirmed that your configuration is in working order, start Snort in daemon mode: `/usr/local/bin/snort -c /etc/snort/snort.conf -i eth1 -g snort -D`.

Run `ps aux | grep snort` and be sure you see something along the lines of

```
root     28293  9.2  8.6  93716 88960 ?        Ss   13:56   0:11 /usr/local/bin/snort -c /etc/snort/snort.conf -i eth1 -g snort -D
```

indicating a running Snort daemon.

Install Aanval[2]

For purposes of brevity, I'll describe the quick install process. That said, I can't stress enough the benefit of reading the complete guide.[3]

Remote Assessment has made Aanval's installation process very straightforward. Default system settings are such that you can quickly get the console up and running and configure various options after installation. Their installation documentation again assumes you have Snort configured, working appropriately and writing to a MySQL database.

Ideally, before installation you need to create a MySQL database for Aanval and give the user the following privileges to the new Aanval database as well as to your snort database if applicable:

`SELECT, INSERT, DELETE, UPDATE, ALTER, DROP, TRUNCATE` (the installation will test these to ensure they are working before installation). Here is a quick checklist for the rest of the steps:

1. Download the latest stable Aanval release and untar/unzip in `/var/www` or a directory of your choice.
2. Point your web-browser at the web directory `/var/www/aanval` and follow the web-browser installation instructions to install Aanval. After following the installation steps, login and follow the console background processing startup steps. If MySQL was properly configured you'll pass through five installation phases with green success indicators along the way.
3. Remember to change the admin account name and password immediately.
4. Follow the Snort Module Configuration steps and use the menu system in the upper-right portion of your Aanval console. Use the Aanval menu and go to the Optional Modules -> Snort IDS Module. You'll need the snort database name, username, password and hostname that Aanval will use to connect to your Snort database. Enter the correct information, and select *Update*. If you've entered the information correctly, the red text will be removed. Click the *Enable Snort* button to activate the module.
5. Consider the use of the Sensor Management Tools (SMT). While not imperative, the SMTs offer additional signature management options and statistics as well as sensor heartbeats and communications.

Again, refer to the above mentioned installation document for further details.

Basic Aanval Usage

Once your console is installed you'll likely want to immediately survey your network. Perhaps the most essential tool Aanval offers is the Go! Search Box. Typically found in the upper right hand corner of your browser pane, the Go! Search Box is unquestionably the portal to Aanval's power. We'll consider three queries in this article. You'll likely use many more once you start using Aanval regularly.

Syntax for a number of queries can be found by clicking *help* next to the Go! Search Box. We'll use three queries `report:malware`, `report:chat` and `report:p2p` to take a look at scenarios that security analysts must battle on a daily basis: spyware in the enterprise and corporate policy violations at the hands of IM and P2P. But first let me offer a breakdown of Snort rule basics.

A quick Snort rule overview

Snort, originally created by Martin Roesch, is powerfully flexible in its ruleset. A visual representation[4] of a Snort rule looks like this:

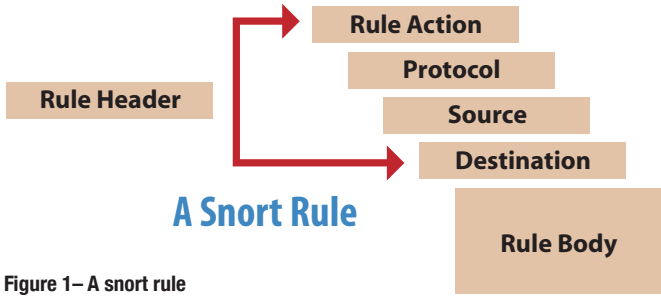

Figure 1– A snort rule

With our visual reference in mind let's use a very simple rule, one that might be used to enforce a "no telnet" policy in your organization.

2 http://www.aanval.com/downloads/aanval_installation_v1.pdf Copyright 2003-2006 Remote Assessment

3 http://www.aanval.com/downloads/aanval_installation_v1.pdf

4 Jay Beale, et al, *Snort Intrusion Detection*, Syngress, chapter 5, p. 151, figure 5.1

```
alert tcp any any > any 23 (msg:"TELNET Viewable
Session"; session:printable;)
```

Using *Roesch's Writing Snort Rules*[5] as our reference, `alert` indicates our rule action, in this case to alert as opposed to log or pass. `any any` represents our source address and port, while > denotes a directional operator which tells Snort to review address/port pairs on ingress traffic. The second `any 23` pair corresponds to our destination address and port. Finally, in parentheses enclosing the rule body, we find the message `(msg)` returned to our console, namely `TELNET Viewable Session`. `session:printable` defines that we should be alerted with data that users can see or type.

So now, armed with the basics, on to the good stuff!

Identifying spyware with Aanval and Snort

In the Go! Search Box I enter `report:malware` and learn that last week's spyware eradication effort must have missed a couple of hosts that had been offline during the sweep, perhaps laptops or PCs with vacationing users. My query results in Figure 2

Keep in mind, that for space's sake, I've shown only a third of the query results, as you will also see source and destination IP addresses and ports as well as multiple pie charts. You may also wish to try `report:trojan` and `report:spyware` as each will yield slightly different results when hunting spyware.

We see that one of our returns is *BLEEDING-EDGE Malware Searchfeed.com Spyware 1.* Matt Jonkman, cofounder of Bleeding Edge Threats, writes many a rule in this tremendous community effort. We will review his rule for the above event.

```
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_
PORTS (msg: "BLEEDING-EDGE Malware Searchfeed.com
Spyware 1"; flow: to_server,established; uricon-
tent:"/rd/Clk.jsp"; reference:url,www.searchfeed.
com; classtype: trojan-activity; sid: 2002296;
rev:2; )
```

Simple in its approach, it breaks down thus:

Our rule action is to alert on `tcp` traffic from our home network to external sites as controlled by the variables `$HOME_NET` and `$EXTERNAL_NET` defined in our snort.conf file. Our protocols are any outbound specifically to defined `HTTP` ports, again managed by the `$HTTP_PORTS` variable in `snort.conf`. The message returned to our console is obviously *BLEEDING-EDGE Malware Searchfeed.com Spyware 1* while the rest of the rule body tells us it's specifically looking for any URI content containing `/rd/Clk.jsp`.

Further querying will show us more. Simply feeding `searchfeed` into the Go! Search Box yields Figure 3

IP addresses have been obfuscated to ensure privacy, but note that you have the option to drill down into the actual payload for further analysis. This especially useful when monitoring more critical events like worm traffic.

This view in Figure 3 will show you some of the payload, but clicking the inverted orange arrow will result in the entire payload, including the hex.

Once you've identified spyware emanating from a particular host you can target it for cleaning in whatever manner your organization employs.

Identifying instant messaging with Aanval and Snort

One of the most prevalent concerns in enterprise computing centers is controlling instant messaging (IM). I'm not writing here to oppose or advocate IM; my interest is purely awareness. However, as a security professional, I cannot stress enough the importance of understanding the risks IM pose to your networks.

In a world befitting of my rule-centric, paranoid brain, let's imagine we work in an environment where IM is simply not allowed, period. Great! That makes our jobs easier. If it shows up in the Aanval console, we know we have an issue.

To see what sort of IM traffic may be flying around your network, type `report:chat` in the search window of your Aanval console and click Go. Depending on your network, you might see a number of different events in the Event Details pane, including generic CHAT rules found in the standard Snort rule set. But, since we're focused on the bleeding edge here, let's zoom in on *BLEEDING-EDGE CHAT Yahoo IM* successful logon. You can simply highlight

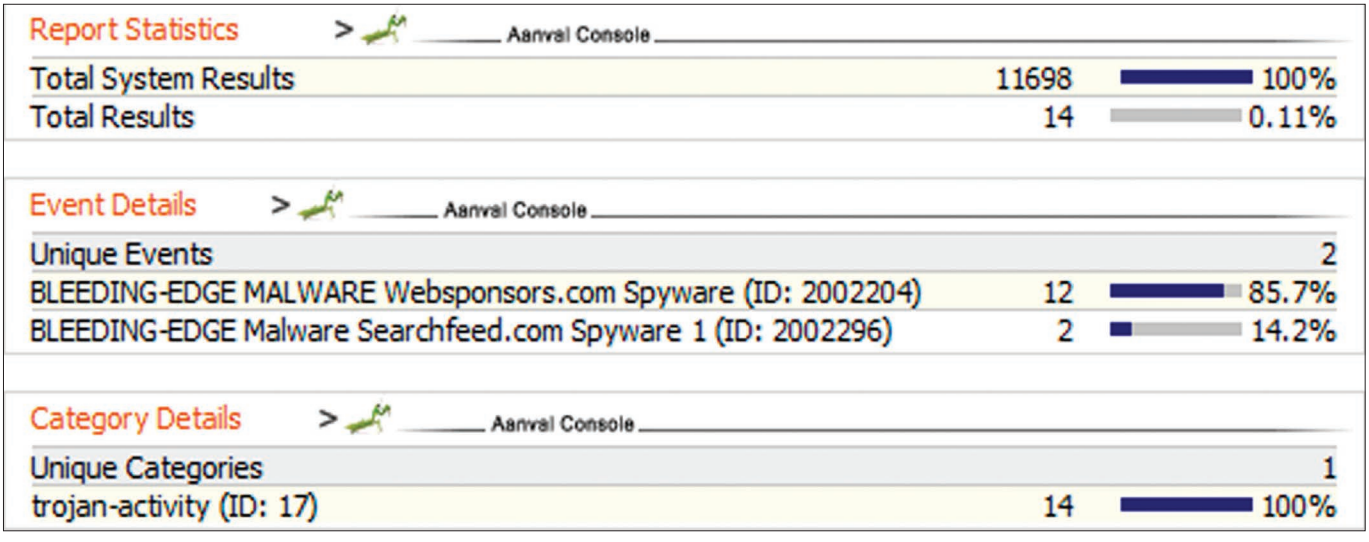

Figure 2 – report:malware query results

5 Martin Roesch, *Writing Snort Rules, How To Write Snort Rules and keep your sanity*, p. 10 http://packetstormsecurity.org/papers/IDS/snort_rules.htm

Figure 3 – Results of feeding "searchfeed" into Go! Search Box.

an event, copy and paste it into the Search box and click Go. The results should look like Figure 4.

Again, the IP addresses have been obfuscated to ensure privacy, but rest assured, the left address is that of the IM server and the right address is the destination host in your network. One critical note on privacy: keep in mind that you can easily intercept IM content, so be very clear about your organization's privacy policies. If you haven't already, immediately implement a logon banner stating that use of your organization's PCs indicates acceptance of monitoring.

I cannot stress enough the importance of understanding the risks IM pose to your networks.

Here's Joel Esler's rule that returned the above alert:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg: "BLEEDING-EDGE CHAT Yahoo IM successful logon"; flow: from_server,established; content:"YMSG"; nocase; depth: 4; content:"|00 01|"; offset: 10; depth: 2; classtype: policy-violation; sid: 2001253; rev:3; )
```

Broken down, the action is to alert tcp traffic from the source address and protocol, controlled by the `$EXTERNAL_NET` variable to the destination address controlled by the `$HOME_NET` variable. We see some new parameters in the rule body that are more involved than our spyware example.

`flow` is use in conjunction with TCP stream reassembly.[6]

The first content reference should make perfect sense: look for `YMSG` and `nocase` makes that search case insensitive.

`offset` is a content rule option modifier. In this case the offset is 10 bytes deep into the payload to avoid searching too early where relevant content may never be found.

`depth` is also a content rule option modifier. To quote Roesch, "It is useful for limiting the pattern match function from performing inefficient searches once the possible search region for a given set of content has been exceeded." So, offset protects the rule from firing too early, and depth too late. These are excellent efficiency options.

`content:"|00 01|";` allows the rule to capture specific bytecode; good for describing complex binary data as hexadecimal numbers.[7]

This rule tightens the search to guarantee an accurate result when a user logs on to Yahoo Messenger. Once identified, your policy enforcement mechanisms can engage to thwart the risk.

Identifying P2P with Aanval and Snort

While spyware and IM signatures may stay fairly stable, you can count on P2P networks changing fairly regularly to avoid detection. Port numbers are the most likely candidate for change so you'll often see Bleeding Edge rules defined with

`alert udp $HOME_NET 1024:65535 -> $EXTERNAL_NET 1024:65535` to capture all traffic above well-known ports.

The Aanval Go! Search Box feature will again allow you a number of options. You could query `report:p2p` for a general search or drop *kazaa, limewire,* or *morpheus* straight in for specific results.

One of the benefits of identifying P2P traffic with Aanval is the two birds, one stone scenario: P2P has obvious risks in and of itself, but

6 Elizabeth Faultersack, *Understanding and Writing Snort Signatures*, Idaho National Engineering and Environmental Laboratory, http://cio.doe.gov/Conferences/Security/Presentations/FaultersackL.pps

7 Martin Roesch, *Writing Snort Rules, How To Write Snort Rules and keep your sanity*, pg 7, http://packetstormsecurity.org/papers/IDS/snort_rules.htm

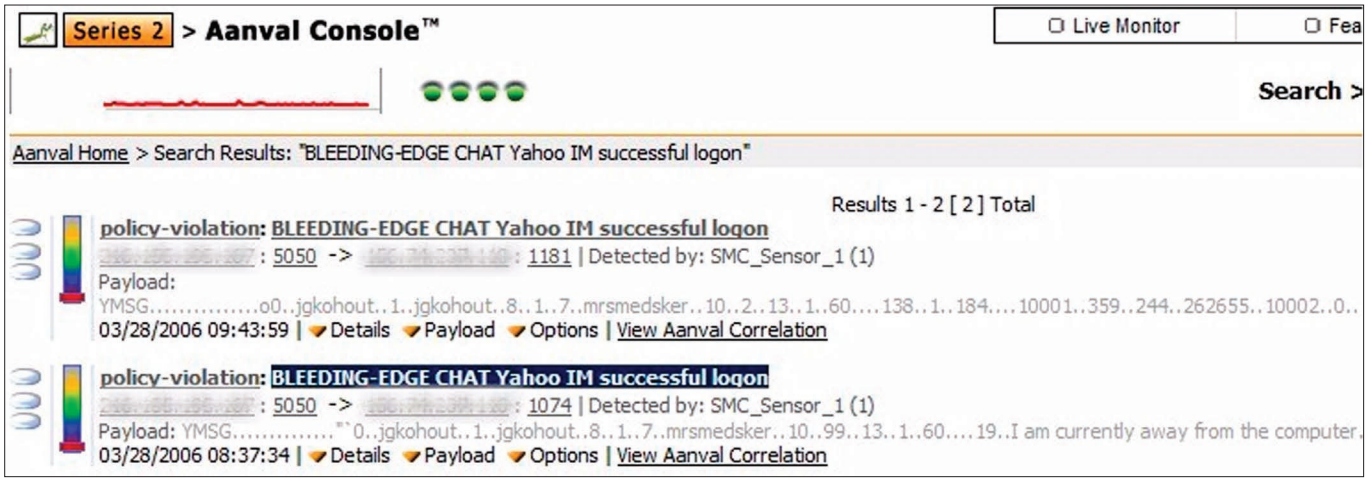

Figure 4 – Results of report:chat query.

its installation on an end-user PC will inevitably result in the installation of spyware included in the installer package.

One last option seen in some Bleeding Edge Threats rules that can be quite useful can be found it this rule by Marcamone:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any
(msg: "BLEEDING-EDGE P2P Ares GET"; flow: estab-
lished; content:"ares"; nocase; pcre:"/(GET |GET
(http|https)\:\/\/[-0-9a-z.]*)\/ares\//i";    refer-
ence:url,www.aresgalaxy.org; classtype: policy-
violation; sid: 2001060; rev:6; )
```

The `pcre` option takes advantage of Perl-compatible regular expressions. Quoting pcre.org, "the PCRE library is a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl 5." This is quite useful in a Snort rule to capture all possible variations of specific content, in this case `ares`.

If you used Patrick Harper's install doc,[8] you're already benefiting from PCRE.

8 http://internetsecurityguru.com/

Summary

This article espouses Aanval, largely from a technical end-user perspective, but most importantly from the perspective of practicing information security. Using Aanval for compliance measurement and policy enforcement will undoubtedly aid you in protecting you network from spyware, IM, and P2P networks.

Use it in good stead and stay vigilant. Until next month…

About the Author

Russ McRee, GCIH, is a security analyst working in the Seattle area. He is a member of ISSA, Pacciso, InfraGard, and CCSA (Cyber Conflict Studies Association). Russ maintains holisticinfosec.org. Contact him at russ@holisticinfosec.org.