CS460 Lab 5 – Intrusion Detection

Due

April 13 on compass

Goal

Use the snort intrusion detection package to analyze traffic and create a signature to identify problem traffic.

Lab Setup

Snort 2.8 is installed on the FC12 snort-linux images on each machine.

The snort source tree is unpacked in Alice's home directory. The snort 2.8 user's manual is in the doc directory of the source tree. In addition, I have several copies of the snort manual for use in the lab.

For full snort information, see http://snort.org.

Important Snort Files

- /var/log/snort – Snort log and alert files. You can use wireshark or tcpdump to look at the captured packets in the log files. The alerts are in ascii.
- /etc/snort/snort.conf - Default configuration files. Controls the built in controls and loads the rule files. Use this to tune the signatures you are scanning for.
- /home/alice/snort-orig.conf – Original version of the snort configuration file. Make a copy of this to adjust how snort runs for you. Use the -c option of snort to pass in your personal copy of the snort configuration file.
- /etc/snort/rules – Signature rule files.

Running Snort

You can run snort in several ways: as a sniffer, as an out-of-band NIDS, run from a file capture, or inline.

Options to run as a sniffer:

- snort -v or snort -dv or snort -dve: Vary the amount of detail captures.
- Snort -dve -l ./log -h 192.168.100.0/24: Specify a log file and the home network. Default home network can also be specified in snort.conf file.

Options to run as NIDS

- snort -dve -c /home/alice/my-snort.conf: Use default log, Run from the snort config file. Still capture packets.

Options to run from a file capture. Use default log. Evaluate signatures against the packets captured in

file “captured-packets.pcap”.

- snort –r captured-packets.pcap -c /home/alice/my-snort.conf

Run in inline mode. You won't be able to do this on the snort VMs. If you are really interested, we can set up a machine to play with inline mode. First you need to set the device to run in transparent bridge mode. See the script at http://www.cs.uiuc.edu/class/sp10/cs460/assignments/setbridge.sh for the commands to set eth0 and eth1 into bridge mode.

Now we need to set the iptables to pass all packets to the queue to be processed by snort

- `modprobe ip_queue` – Load the kernel support for QUEUES. The QUEUE is a means to pass packets from kernel to user space.
- iptables -I FORWARD -o br0 -j QUEUE

No traffic will pass until we turn on snort

- snort -c /home/alice/my-snort.conf

The bridging commands were gleaned from the more general rc.firewall script distributed with the honey net project (http://www.honeynet.org/tools/dcontrol/rc.firewall). This script also supports a L3 routed mode which they call NAT mode. The benefit of operating in a bridge or L2 mode is that you can insert the snort box without adjusting your address distribution or routing logic. From the honeynet perspective, it makes it harder for the attacker to notice that you might have an interception point.

Hand in requirements

Review the rules and update the set of loaded rules in your copy of the snort.conf file. In your writeup describe which rules you activated and why.

Run nmap through or past the snort box. Try using ftp to get some interesting system files. Describe what if any alerts you get in these cases.

Use snort's “-r” option to replay the logs in the /home/alice/logs directory: snort –r <packet file name> -c /home/alice/my-snort.conf. For each log file do the following:

- Find the rules that caused the alerts. In some cases there will be many, many alerts, but there should only be a few classes of alerts. For each file research five alerts that represent different classes of attacks. If there are less than five classes of alerts in that file, research an instance from all classes of alerts.
- In your writeup describe the meaning of the alerts.
- Speculate on whether the alerts were due to real attacks or false positives. If you believe the alerts were due to real malicious behavior, how would you address the problem?
- More details on the network where the logs were taken will be discussed in class and posted to the newsgroup.

Write a signature that differentiates good and bad magic8 packets. Include an alert that is generated from this signature. In your writeup include the signature and your description of the signature. As the magic8 exploit writer, how would you adjust your attack to avoid this signature? Ultimately will the attacker or defender win in a signature/exploit adjustment war?