Improving Network Security Using Ntop

Luca Deri[1,2] and Stefano Suin[2]
Finsiel S.p.A[1]., Via Matteucci 34/b, Pisa, Italy.
Centro Serra[2], University of Pisa, University of Pisa, Lungarno Pacinotti 43, Pisa, Italy.

Ntop has been originally designed as an open source, web-based traffic measurement and monitoring application, easy to deploy by network administrators. As ntop has been used for analysing traffic patters, some users requestes some facilities for classifying traffic hence recognising specific attacks. In order to address these requests, the authors decided to extend ntop adding an embedded NDIS (Network Intrusion Detection System). What makes ntop NIDS unique from other available NDIS is its knowledge of the monitored network. While capturing packets, ntop learns network topology and hosts relationships (i.e. routers, DNS, networks) and stores this information in a network knowledge database. This knowledge is dynamic and not specified at ntop start-up by means of configuration files. For instance, if host X successfully routes packets for host Y, then ntop assumes that X is a router for host Y. Similarly, if host K sends packets with different source IP addresses and a single MAC (Media Access Control) address, then K has enabled multihoming support. Ntop knowledge database is updated as new packets are captured and is not static whatsoever.

Network knowledge simplifies the task of specifying rules for defining burglar alarms. The following example justifies this statement. In a network, routers are the only hosts entitled to send ICMP (Internet Control Message Protocol) redirect. If host X receives an ICMP redirect from host Y and host Y is not used by X as router, then either Y is misconfigured or Y is trying to capture X traffic by asking X to route its traffic through it. Without dynamic network knowledge, a user the XYZ IDS has to specify the IP addresses of all the known/used routers and then define one rule for each router. Ntop instead, can do the same job using exactly one rule (burglar alarm): `"icmp route-advertisement ICMP_ROUTERADVERT !defaultrouter/any alarm"` that means "emit an alarm labelled route-advertisement whenever host X receives an ICMP route advertisement from host Y and Y is not a router used previously by X. It is worth to note that the previous rule does not specify the IP address of X and Y, nor the number of possible hosts/routers. Instead, the rule specifies a fact that will always hold regardless of the IP address, network topology, and media type. As shown in this example, network knowledge exploitation allows an IDS to:

- define fewer rules for a network event (advantage: shorter packet processing time);
- specify rules independently of the network where the IDS is used (advantages: less error prone, network-independent rules, rule portability across networks with both dynamic -by means of DHCP (Dynamic Host Configuration Protocol)- and static addressing).

The network knowledge disadvantage is that a rule is not effective if the knowledge database is small and if an attacker is able to fool ntop. In this case ntop will make some wrong assumptions and then will not be able to recognise specified traffic patterns. Nevertheless, as specified before, ntop is rather conservative about network knowledge hence the risk of fooling ntop should be rather low.

Another important feature of ntop NIDS is the embedded event correlation engine. When a rule is matched by the current traffic an event (either informational, warning, or alarm) is emitted. Usually these events are sent to the management console, if present, or delivered to network administrator by various means including email, and SMS (Short Messaging System). Unfortunately if a rule matches, it is likely that the same rule will match again in the near future because if the problem is not promptly solved then it will happen again. Suppose for instance that host X is attacking host Y with a DOS (Denial of Service) attack. X will probably send a large number of packets with the TCP SYN flag set to the target host. If ntop contains a rule that says "if an host sends more that 50 packets in 5 seconds with the SYN flag set to a target host then emit an alarm", the network administrator will likely receive an alarm concerning the attack once every 5 seconds until the attack is over. Be-

side the fact that the network administrator might be rather annoyed by receiving all these alarms, the IDS will generate itself a large amount of traffic that could eventually turn the IDS into a DOS tool because it will flood the network administrator host with a storm of messages causing as a waterfall effect.
The event correlator can also be used for recognising attacks by monitoring activities that do not complete successfully. For instance suppose that an intruder is trying to learn about the local network before to begin the attack. Very likely, the attacker will scan the ports of the target host or will try some operations that will eventually fail. For instance, the attacker will attempt to connect to non open ports on the target host, or will ping hosts that are either down or inexistent. As these operations do not succeed, either the attacker will receive an error message or a missing response. In general these are probe events, so they identify problems only if their rate is higher than a specified threshold in a given amount of time. The event correlator can be profitably used for recognising these problems. In fact, the network administrator can define two rules: “emit an event of type A whenever you see a TCP packet with the SYN flag set unless this event is cleared before 60 seconds” and “clear event A if you see a TCP packet with the ACK flag set on the connection that generated event A”. The event correlator first sees the event A, and it waits 60 seconds before sending the event to the network administrator. If within 60 seconds the event A is cleared, then the event is deleted. Instead if the timeout expires, the event is finally delivered to the network administrator because it has not been cleared within the expected timeframe. Without the event correlator it would be more difficult -if possible at all-to detect the above attacks. The use of an extern correlator has the disadvantage that both the NIDS and the correlator need to be kept in sync, and that the events emitted by the NIDS can potentially generate a lot of traffic and will likely be filtered out by the correlator.

This paper presented the phylosophy that is behind the design of ntop NIDS and has show how it differenciates from similar efforts. Due to space constraints, we are not able to give an accurate rules description and explain how ntop works internally. Readers interested in learing more about ntop and how to write NIDS rules can visit the ntop home page, http://www.ntop.org/.

Ntop for both Unix and Win32 is distributed under the GPL2 licence and can be downloaded free of charge from both the ntop home page and other mirrors on the Internet. Some Unix distributions including but not limited to FreeBSD and Linux, come with ntop preinstalled.