Protocol Anomaly Detection in Network-based IDSs

Erwan Lemonnier - Defcom

28th June 2001

Abstract

This white paper aims at briefly describing the technologies currently used in filter design in Network-based Intrusion Detection System (NIDS). We will consider the advantages and drawbacks of using signature filters versus anomaly filters, and more particularly protocol anomaly filters.

This is the result of research work done at Defcom Sweden, Stockholm.

1 Common NIDS architecture

Intrusion Detection Systems (IDSs) are the computer equivalent of office burglar alarms: they monitor a computer system in an attempt to detect intrusions. By intrusion, we mean activities that are threatening the security model implemented in the monitored computer system. IDSs come in two flavors: Network-based IDS (NIDS) and Host-based IDS (HIDS). NIDSs are specialized into monitoring network traffic, while HIDS are programs monitoring specific host computers. We will focus here on NIDSs.

A simple model of NIDS consists usually in one or more network-based sensors, running filters, which are generating an alert flow gathered at some monitoring central, as shown on figure 1. These network-based sensors are made of a network interface put in promiscuous mode to sniff all the traffic going through the local network segment, and of a TCP/IP stack processing the received packets before giving them to the filters.

NIDS filters are small programs analyzing the data flow produced by network-based sensors in order to detect patterns of suspicious activity. There are mainly two technics currently available for performing this pattern recognition: signature detection, and anomaly detection. We will thereafter introduce the advantages of a sub-case of anomaly detection: protocol anomaly detection.

2 Signature filters

Signature filters are the simplest and most widespread type of filter. A signature filter basically looks for one specific signature in the data flow provided by a sensor. A network attack does indeed usually possess some kind of signature that identifies it.

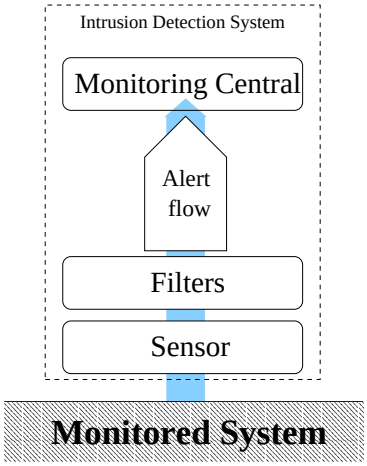

Figure 1: Simple NIDS architecture

This signature often consists in one or more specific binary pattern found in a given network packet[1]. This signature can be described as a boolean relation called rule.

Although simple to implement, signature filters have a number of limitations. They are able to recognize an attack only when they 'know' a signature for this attack, and thus require continuous updates of their signature database as well as continuous research work to analyse new attacks and find their signatures. Moreover, a slight change in the attack scenario may be enough to alter the attack signature and thus fool a signature filter. They are consequently vulnerable to polymorphic attacks and other evasion technics which are expected to grow in the near future.

3 Statistical Anomaly filter

A statistical anomaly filter is designed to monitor a system and detect statistically exceptional events. The idea is to build a model of a system's normal behavior under safe conditions, and then to periodically compare the system's behavior against this model. If the difference exceeds some limit, an alert is generated.

In a network based context, the behavior monitored could be a user's activity over the network, or a protocol traffic. The model for a normal behavior would be obtained by sampling the corresponding traffic under a period of time representative of safe activity. The same parameters would then be sampled again on a regular basis, and the result compared with the reference model.

How evolutive a reference model should be is a source of dilemma. Systems' behaviors tend indeed to change with time, and statistical anomaly filters should consequently be designed to allow regular updates of their reference model. On the other hand, if this model is updated too often, an intruder could spread his activity over a pe-

[1]Example: an http get request to msadc.dll, or to an url containing unicode characters for '../'

riod of time long enough to let the filter 'learn' his behavior and accept it in its model as a normal behavior.

4 Protocol Anomaly filter

Protocol anomaly filter can be considered as a special case of statistical anomaly filter to which specific protocol knowledge has been included.

Theory

We call protocol anomaly filter a filter looking for protocol misusage. 'Protocol' should here be interpreted as any official set of rules describing the interaction between elements in a computer system. A protocol anomaly filter is consequently designed to analyze specifically one protocol, and requires to define a model of this protocol's 'normal' usage. Protocols always have a theoretical usage, corresponding to their official description in documents such as RFCs [2], but experience shows that they are seldom implemented and used in complete accordance with these official descriptions. A model for a protocol's 'normal' usage can thus be defined as the superposition of the official and practical area of usage of this protocol. Any use of this protocol outside these intersected areas can be considered as a protocol anomaly. Figure 2 illustrates this situation through the two overlapping colored circles representing practical and official protocol usage.

The interesting point is that experience shows that 90% of the attacks can be considered as protocol usage anomalies. The reason of it lays in the fact that most of the attacks are exploiting breaches in badly defined areas of protocols, in which special cases have been neglected in the protocol standard itself as well as in its implementations.

Protocol anomaly filters are thus able to detect all attacks that are using protocols outside of their normal usage area, as shown on figure 2, which especially includes new attacks that may not yet have been registered by computer security authorities. This ability of detecting new attacks, added to the fact that they don't require signature database updates and have the same long lifetime as the protocol they are monitoring, makes the superiority of protocol anomaly filters on signature filters.

Practice

When running above a network based sensor, anomaly filters would disassemble the data packets for each network protocol, and check if they are built in compliance with the protocol standards, as described in RFCs or equivalents. Practice shows however that protocols are seldom implemented according to their standards, and anomaly filters should thus be designed in a flexible way, in order to fit not only to the official usage of a protocol, but mainly to a given model of a protocol's 'normal' usage. Building such a model requires to analyze common protocol implementations in practice, in order to define the limits of what is officially and unofficially the standard for this protocol.

[2] Request For Comments are the official written definitions of the protocols and policies of the Internet.

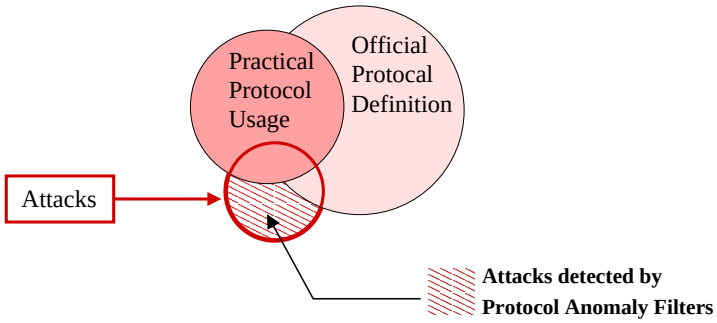

Figure 2: Attacks detected by Protocol Anomaly Filters.

An example of protocol anomaly could be extra large arguments passed within a protocol command. Though extra large arguments are often not considered in official protocol standards as explicitly illicit, they should be considered in practice as a suspicious sign, since they are not justified by normal usage and may reflect an attempt to exploit buffer overflows or denial of service (ex: giving an exceptionally large username to a telnet server at the authentication stage causes the server to crash...).

5 Pros and Cons

Protocol anomaly filters are theoretically more performant than signature filters. They can potentially catch most of the attacks, including new and unknown ones, without requiring attack-dependent knowledge. While longer to develop than signature filters, they however have a much longer lifetime and will not require regular updates.

Yet, interpreting alerts generated by protocol anomaly filters is a difficult task, since they do not provide clear information about the nature of the threat. They should therefore be monitored by experienced personal. Besides, these filters are not able to detect the few attacks that can not be considered as protocol anomalies. Hence it is advised to use them in collaboration with signature filters.