

Rules definition for anomaly based intrusion detection

v1.1

©2002-2003 by Lubomir Nistor

Introduction

Intrusion detection systems (IDS) are one of the fastest growing technologies within the security space. Unfortunately, many companies find it hard to put in use due to complexity of deployment and or lack of information about its possible use. This document should help security experts, integrators or end-customers to utilize their IDS system to its limits or to fit the expectations required by company.

The market is currently filled mostly by rule-based IDS solutions aiming at detecting already known attacks by analysing traffic flow and looking for known signatures. This fact requires such IDS to be under constant construction updating and modifying attack signatures and requiring to pay considerable financial amount for support.

On the other hand it is possible to use anomaly based IDS solutions detecting not just known attacks but also unknown attacks and informing network engineers about possible network problems or helping them to troubleshoot them.

There is no clear answer which solution is better as they have their advantages and disadvantages, but there is a possibility to put the rule-based IDS solutions in use as if they were anomaly based. This document describes possible ways of doing that by modifying the signatures. All the examples and solutions are based upon Snort IDS that is open-source solution freely available and well established on the market. Although this solution is open-source there are many companies offering support or even appliance or turn-key solutions.

Data flow identification

The whole network security is based upon allowed suspicious and not allowed traffic. Identifying what packets fall under which category is the most difficult task of all. Very often even the system owners don't know what kind of traffic they can expect and therefore it is necessary for network security professional to know the most used and known protocols and networking systems.

As in every analysis stage of project the more information analyst can gather the better. More detailed information about data flow helps to specify exactly the traffic expected by end systems.

Information expected:

- Source, Destination IP/network
- Network protocol (IP, ICMP,..)
- Application protocol (distinguished by port nr.)
- Protocol options
- Content (size, type, characteristic strings,..)
- Other features (sequence nr., TCP flags, TTL,..)

Defining ALLOWED category

Here fall all the packets complying with clearly with requirements of system owners. It is necessary to ensure that also the content is acceptable as communication is not just an IP address and a port. It is possible to search the content for specific values or check the content size in order to determine the packet's category.

Defining SUSPICIOUS category

By reviewing required services there are many options that are not required or used by end systems. Although according to port and IP address they are allowed to pass the packet filter or firewall but this protocol option is not expected or not supported by end system. As an example in HTTP protocol if website doesn't use POST method and such method is used in the packet it surely should raise an alert or at least inform about this action.

Defining NOT ALLOWED category

All the traffic that doesn't fall under any other category should be put in this category. Packets that are not expected by any services, systems or network devices fall under this category. Even packets with allowed services that have invalid destination or packets with correct destination but invalid services should be in this category.

Solution 1

(pass allowed, alert the rest)

This solution is based upon allowing traffic expected and issuing general warning about the rest. Information taken from data flow identification

Configuration example

```
#web server
Pass TCP $WEB_SERVER 80 <> any 1024:
#admin access
Pass TCP $ADMIN_IP 1024: <> $WEB_SERVER 22
#DNS
Pass UDP $WEB_SERVER 1024: <> $DNS_SERVER 53
#alert rules
ALERT IP any any <> any any (msg: "not allowed traffic");)
```

Advantages

Can catch any not standard traffic incl. very slow portscans and directed probes
Provides also troubleshooting capabilities for network engineers

Disadvantages

Too many log entries under the same category
Hard to filter according to message or priority

Implementation possibilities

- High security areas with low “not-defined” traffic
- Areas with no unexpected external or user traffic
- Well defined firewalled and filtered environment
- IDS research, honeypot systems

Solution 2

(alert if any not expected services)

It makes no use to analyze traffic that no server is listening for and therefore this solution is only analyzing the contents of traffic that is directed to existing services. By using protocol definitions it is possible to distinguish between various options or states in communication and define rules that alert in case of unexpected protocol states.

As an example HTTP traffic has a POST state where a client is sending data to the server (can be binary data or options for the website). If none of the webserver's pages are using such protocol option it is suspicious to see such traffic going to the webserver.

Another example is with TCP flags, where by definition of the protocol a packet doesn't contain all the flags. There is a scan technique called "Xmas tree scan" where all these options are in the packet and by defining alert rule like this SNORT is able to detect it.

Configuration example

```
#web server
AERTs TCP $WEB_SERVER 80 <> any 1024: (contain: "POST"; msg: "HTTP
POST is being used")
#admin access
Pass TCP $ADMIN_IP 1024: <> $WEB_SERVER 22 (flags: SFRPA ; msg
"possible Xmas tree scan");
#DNS
Pass UDP $WEB_SERVER 1024: <> $DNS_SERVER 53
#alert rules
ALERT IP any any <> any any (msg: "not allowed traffic");
```

Advantages

- Less log entries for not expected traffic
- More relevant alerts with different prioritization and detail
- Possible log filtering according priority and SID

Disadvantages

- Network scans and probing can't be detected easily
- Update of rulebase necessary after introducing new or modifying old services in the network
- Rules have to be done in more detail

Implementation possibilities

- DMZ or filtered environments with well defined services
- Medium level security networks with defined active services

Alternatives

Trend analyzers

Each entity (user or system) has it's trend in communicating or generating network traffic. Email server generates a lot of email traffic or user network generates a lot of HTTP traffic and by identifying such trends it is possible to observe a trend in general network traffic coming through one point in the network.

[Lancope](#) uses flow-based anomaly detection in their [Stealthwatch](#) product. This program "characterizes and tracks network activities to differentiate abnormal network behavior from normal".

Packet analyzers

Many packet analyzers try to verify if network traffic complies with RFC standards or generally accepted implementations. By detecting packets not complying with protocol standard or communication trend they raise the alert.

One good example is arpswatch, which monitors ARP traffic and when detecting a change in MAC <-> IP relation it alerts the administrator. General communication standard of ARP traffic is that IP address doesn't change it's MAC address in a static network (although there are some exceptions).

Another good example is stateful packet inspection, where state of communication is being monitored and in case of any deviation from this state alert would be raised (or the packet dropped)

In [PGP Personal edition](#) there's a PGPNet personal IDS system integrated that analyses basic protocol violations and reports them.

Protocol analysis for link state routing protocols:

<http://www.cs.ucsb.edu/~rsg/Routing/references/qu98statistical.pdf>

Statistical analyzers

Each network traffic has it's key identifiers either qualitative or quantitative. By monitoring these identifiers an IDS is able to detect anomalous traffic and report it. For example it is very suspicious to see increase of ICMP traffic from 1KB/s to 10MB/s or one IP address sending SYN packets to every port.

Statistical Packet Anomaly Detection Engine for SNORT:

<http://www.silicondefense.com/software/spice/index.htm>

Another product is a C.P.M.A.D in [Checkpoint Firewall 1](#) that according to log entries from the firewall engine produces alerts for common attacks or scans.

Conclusion

During the time of Economic recession it is necessary to save as much finance as possible in order to stay competitive. According to all my experience this is a very precious feature that not many managers have, but I hope that evolution is still affecting our lives and only the best would survive on the market.

These best IT managers and experts see the bright future in open-source solutions and would put information provided in this document in good use.

For all the rest I hope this document helps you to utilize your IDS system to increase the security level as your management expects it.

And don't forget it's not the IDS that secures your company but the people who manage it.

Appendix A. W32.Blaster detection

There are many variants of anomaly tracking deployment and one of them is monitoring network performance and protocol distribution. In my current consulting engagement in space industry I was tasked to deploy a network performance monitoring system. With a lot of good experience with open-source technologies I decided to use Ntop in order to produce nice reports and easily manageable interface. After several weeks spent reducing the unnecessary traffic I was glad that the protocol distribution was as normal as could be.

But recently there started to be increased activity on windows ports that forced me to have a look at network logs. There I could see many IPs communicating on port 135 to the outside. Each machine sent SYN packets to a dozen outside IP addresses in a row. For a scan the ranges seen were too small and only on that port. I also know that from outside there are many scans for “NULL session” bug opened machines but it was not normal from the inside.

This led to defining it as anomaly.

Due to communication from other company from the same industry we were alerted that there is a new virus spreading called W32.Blaster. After verifying the symptoms of the virus it was clear that we had some infections too. Thanks to the IDS system I was able to identify specific machines and their owners and first level support was able to clean them and update the virus definition tables to prevent it.

As you can see on this practical example it is possible to detect Trojan horses or worms in the early stage before the CEO runs down to you and state the warning of virus attack from CNN.

Although the worm was not well designed a similar infection can be detected by monitoring network performance. Due to cases of very smart worms it may have stealth techniques and infect machines more slowly to avoid being detected by performance monitoring, but it would be still detected by anomaly IDS identifying non-standard network transmissions.

This is a major advantage from any other IDS or firewalls except good security design (maybe I forgot to mention that my network was not infected if I don't count users who brought it from home but even those computers didn't spread it anywhere).

Appendix B. Anomaly conditions in LAN

Here are some examples showing anomalies in a LAN environment:
 Their use can be helpful in resolving packets captured or by defining heterogenous
 IDS systems (rule-based + anomaly detection)

Condition	Description
<ul style="list-style-type: none"> • SYN packet • SRC=local LAN • TTL=[Standard TTL - 1] 	There is a router in the network(possibly with NAT) Standard TTL is 256/128/64
<ul style="list-style-type: none"> • Src port = 21 • Packet outgoing 	A FTP server is in the network
<ul style="list-style-type: none"> • Port 135-139 • Going through gateway 	Either a Trojan horse/worm/virus or a wrong configuration of a windows machine
<ul style="list-style-type: none"> • TTL=1 • DST=local LAN 	Somebody is tracerouting a IP in your network
<ul style="list-style-type: none"> • TTL=1 • SRC=local LAN 	Somebody is tracerouting from your network (finding a way out)
<ul style="list-style-type: none"> • 	