# Network Defense and Intrusion Detection

Andrew Bates

## Abstract

*Network Intrusion Detection Systems (NIDS) have been around for many years. The founding principle of many designs is that a good NIDs will be aware of normal traffic (tcp traffic to port 25 for an email server, for instance) and be able to alert anything that is not classified as normal. However, many enterprise networks are far to large and complex to easily classify all systems and traffic. Further, greater risk is being introduced by an attack style known as the Advanced Persistent Threat (APT). A new form of Network Anomaly Detection needs to be studied and the anomaly detection must take place as close to the host as possible in order to detect malicious traffic even within one's own network. Intelligent sensors should be placed as close to hosts as possible, and perhaps, even on the physical host as a transparent virtual device. This placement should allow much more granular learning of traffic patterns in order to increase alert confidence.*

## 1. Introduction

Defending computers and computer networks has never been an easy task. For every mitigation technique learned a new means of fooling that technique soon follows. File audit programs using weak hashes (such as MD5) can be coerced into reporting no file changes when in fact dramatic changes have taken place[1]. Network Intrusion Detection Sensors (IDS) can be circumvented by using simple encoding algorithms to avoid shell code pattern matches in the sensors. Covert communication channels can even be set up using seemingly innocuous protocols like ICMP [2]. Many defense and intrusion detection schemes are quite good at detecting (and even preventing) certain kinds of attacks. Distributed denial of service attacks and known virus traffic can be detected relatively easily. However, if an attack takes place over a great deal of time (greater then 3 months) and uses a very low volume communication channel, it can be almost impossible to detect. If such an attack uses a zero day exploit [3] of some service this can lead to a disastrous outcome. Likewise, an attack may not use a vulnerability at all but may simply convince a legitimate user to install and run software that initiates an attack.

While detecting and reacting to various computer based threats is quite difficult, there are ways to protect assets and mitigate certain risks. Using firewalls and implementing compartmentalized access, for instance, can limit the extent of damage an attack might cause. Implementing good software updates can also prevent attacks using well known exploits. Also, maintaining a good intrusion detection infrastructure (both host based and network based) can help to alert that an event may be taking place. However, implementing good practices in network and software design is no longer enough. Advanced attacks today can be so convincing as to fool even the most security conscious individual. Pattern based detection schemes, used by anti-virus software as well as simple intrusion detection sensors, are always a step behind attacks. A signature of an attack must already be known in order to build a rule that can detect it. Yet another limitation is that pattern based intrusion detection sensors at the perimeter of very large networks can produce more false positives then correctly identified malicious traffic. One approach to reducing the number of false positives would be to push the profiling and detection system back into the network as close to the host as possible.

This paper will propose a scheme to push the intrusion detection system back as far as possible, even as far as the physical resource originating the network traffic. Rather then using simple pattern based detection mechanisms this paper will describe a framework to be used to profile normal behavior and use that to detect abnormal behavior.

## 2. Intrusion Detection

Pattern based intrusion detection tends to generate a high volume of data. Many of the alerts generated by pattern based intrusion detection sensors can be classified as being false positives. In this case, false positive may be

misleading. Network traffic indeed caused an alert to match, however that traffic may be harmless and may simply be noise generated by legitimate applications. The popular open source IDS software "Snort" is frequently used in even large-scale intrusion detection infrastructures. A Snort sensor with all the vendor supplied rules turned on will generate a tremendous number of alerts. A recent snapshot of data from a small home network with fewer then 5 workstations recorded several hundred alerts in only about 24 hours. Assuming this number has a linear relationship between number of network host and number of alerts, the number of alerts would increase to several hundred thousand alerts for a network of only ten thousand hosts. While techniques do exist to reduce the magnitude of alerts [4[5] for very large networks harmless alerts can quickly become unmanageable.

What is demonstrated here is installing an IDS sensor with default rules. It is ill-advised to simply install an intrusion detection sensor and blindly apply rules. Knowledge about the behavior of the hosts involved must be acquired and the rule-set then tuned to reflect normal versus abnormal behavior. For small networks this task is relatively easy to do. A single sensor could be set to alert on traffic that does not fit the normal pattern of use for that network. For medium sized networks this is also not entirely difficult to accomplish. Sensors can be placed in areas such that groups of system sharing common activity (a human resources department, for instance) are monitored for anomalous activity.

A problem arises for larger networks and organizations. In large organizations resources are often decentralized and spread over geographic areas. A single department may have employees and resources spanning many cities or even countries. Intrusion detection sensors may be deployed to company campuses, but a security professional would find it difficult to build and maintain a rule-set that defines normal activity for a campus where all activity is normal.

Given the potential for a large number of false positives using basic pattern based detection rules it becomes clear that detecting very low volume traffic (such as traffic during an APT style attack) is nearly impossible due to the high noise levels. Reducing the amount of noise in an IDS is certainly a start. Reducing the number of alerts for harmless traffic will allow security professionals to focus on a smaller set of alerts. However, the problem remains on just how to do that. In some cases the number of security professionals assigned to this task is so low that the workload is simply beyond their ability to complete. Likewise, some security tools designed to help with reduction of data are far to expensive for some small companies. Basic tools need to be built that can effectively profile behavior on a system by system basis and then apply rules based on this profiled data. Pairing this with a scoring model to indicate a level of suspiciousness to some traffic may automatically reduce the number of alerts that need to be investigated.

Often, IDS implementations are placed only at points in the network where there is a demarcation point. For instance, sensors may be placed at the connection between the internal network and the DMZ as well as within the DMZ at the egress to the public network. Likewise, sensors may be placed at connection to customer networks as well as in data centers. While this approach can produce a good sample of data, the processing power and equipment required to monitor and profile this magnitude of data (presumably these links are high bandwidth) can become quite expensive. Since most an alternative approach may be necessary to come up with an effected detection scheme. The premise behind pattern based detection is that normal behavior is known and can be documented. Any behavior outside that norm can be alerted and researched. However, as demonstrated it may not be feasible to document or determine normal behavior. This is especially true when the detection system is located at the perimeter of the network.

One approach to solving this problem would be to leverage excess CPU utilization on systems as close to the user or server as possibly. With virtualization technology today it may even be possible to establish a virtual intrusion detection system directly on the individual hosts. The hyper-visor could be configured to allow the IDS access to the hardware network adapter and then provide a virtual bridge available only to the IDS and main operating system. If this were set up in such a way as to be transparent to the user then the illusion of operating directly on hardware could be achieved while collecting data directly on the host. Figure 1 displays this concept graphically. While this will reduce the amount of traffic a sensor must analyze, the problem still exists of determining what is normal versus abnormal behavior. We must find an automated way of finding normal behavior since it is not feasible to manually document every system's expected behavior on a large network.

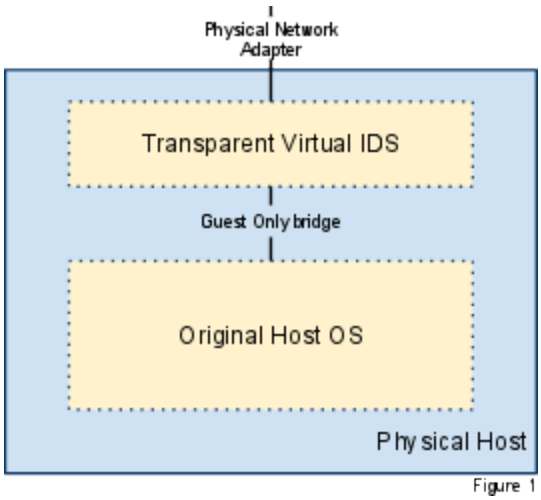

Figure 1

Placing the IDS directly on a physical host would allow that IDS to learn the pattern of behavior for that single host. If a means of profiling the traffic, possibly heuristic or some form of artificial intelligence, could be developed then each system's individual sensor could learn normal traffic patterns and alert on the abnormal traffic, even of a very small scale. A number of key attributes should be identified to be used for the heuristics engine to profile. The following list illustrates some attributes that could be used to profile a system's traffic:

1. Distribution of packet sizes
2. IP, protocol and port combinations including groupings by common services (tcp port 80, for instance)
3. Traffic volume relationships with time of day and day of week
4. Frequency of flows between various geographic areas (could be grouped by Regional Internet Registry)
5. Frequency of connectivity various domains and hosts within top level domains (number of connections to .com domains rather then .ru domains, for instance).

Systems with well known tasks could have an even more granular set of attributes. For instance, systems known to be email relays could track the relationship of external sources to volume of internal emails. Web servers could be profiled by average size of pages served up and by average number of transactions to back-end systems.

## 3. Further Work

This paper simply proposes an idea and expands on speculation. Further work should involve gathering more concrete data including information on feasibility of deployment in an enterprise as well as confidence of detecting anomalies using learning techniques. Once these data have been collected a test bed will need to be built to simulate a realistic environment. Control systems should be built that use standard tools for management and produce a sample of standard network traffic (http, email, etc.) Other systems should be built and profiled using simple pattern based intrusion detection. Finally, the host based virtual intrusion detection system should be built and tested. Tests should include custom virus code, software that builds atypical connections (port 22 traffic not using ssh, for instance) and other attack type traffic. Comparisons should then be made among the control data, the pattern based IDS data as well as the host based VIDS data.

## 4. Conclusions

Advanced Persistent Threats are nothing more then network security threats that have existed for decades. The major aspect of APT style attacks is that those attacks can be low volume and targeted at a narrow set of hosts or people. Standard pattern based intrusion detection is not well suited for detecting APT style attacks since the attack may not use known exploits or any exploits at all. One way of detecting APT style attacks is to establish what

"normal" is for a given host and then alert on anything that isn't "normal". One way of achieving this goal is to place a transparent virtual intrusion detection sensor on the physical host that is a transparent bridge between the physical network and the host OS. This design would allow the VIDS to learn normal behaviour and then alert anything that is classified as abnormal. This design should be testable in a controlled environment and the effectiveness should be easily observed.

## 5. References


[1] Kaminsky, D, "MD5 To Be Considered Harmful Someday," *http://www.doxpara.com/md5_someday.pdf*, December 2004

[2] Lakhoua, M. B., "Bypassing Firewalls using ICMP Tunnel", *https://www.infosecisland.com/blogview/9294-Bypassing-Firewalls-Using-ICMP-Tunnel.html*, November 2010

[3] Various, "Zero-day attack", *http://en.wikipedia.org/wiki/Zero-day_attack*, December 2010

[4] Gaonjur, P., Tarapore, N.S., Pukale, S.G. and Dhore, M.L., "*Using Neuro-Fuzzy Techniques to Reduce False Alerts in IDS*," ICON 2008. 16th IEEE International Conference on Networks, New Delhi, pp1-6, 2008

[5] Wang, Y. "*Statistical Techniques for Network Security*", Hershey, PA, Information Science Reference, pp172-205, 2009