Strategies to Reduce False Positives and False Negatives in NIDS

Network-based intrusion detection systems (NIDS) perform in-depth packet analysis in order to enumerate attackers who are attempting to expose network and service vulnerabilities. NIDS devices can also aid in identifying misuse patterns and gathering forensic data. By examining network traffic in real time, NIDS devices can alert users to possible attacks and/or take predefined responsive actions to help mitigate the threat. By providing an additional layer of protection above and beyond access control devices such as a firewall, NIDSs can be a valuable addition to the security arsenal. However, network intrusion detection has been criticized for its propensity to generate a perceived large amount of false positives and false negatives. Effective NIDS device management can appreciably reduce these reporting inaccuracies.

This article is the first of a two-part series that will offer an overview of network-based intrusion detection and false reports. This installment will offer a brief overview of NIDS devices and will examine how and why false reports take place. The upcoming second article in this series will offer strategies and techniques for reducing false positive and false negative alarms, thereby increasing the efficiency and efficacy of NIDSs.

False Positives or False Alarms

The term false positive is a broad and somewhat vague term that describes a situation in which an NIDS device trigger an alarm in a when there is malicious activity or attack occurring. Other common terms used to describe this condition are "false alarms" and "benign trigger". False alarm is the better term to describe this behavior since "false positive" gives the impression that IDS technology itself is fundamentally flawed and benign trigger gives the impression that there is no possibility for a true false positive to exist. Here I will use the term false alarm to describe the general condition of an alarm being generated without a true security related event. False alarms are the Internet security equivalents of the boy who cried wolf. They are problematic because by triggering unjustified alerts, they diminish the value and urgency of real alerts.

Beyond False Positives: Categories of False Alarms

This discussion has thus far been phrased in terms of false positives and false negatives; however, false alarms can be subdivided into several more meaningful and specific categories. Common categories into which false alarms can be divided include:

- **Reactionary Traffic alarms:** Traffic that is caused by another network event, often non malicious. An example of this would be a NIDS device triggering an ICMP flood alarm when it is really several destination unreachable packets caused by equipment failure somewhere in the Internet cloud.
- **Equipment-related alarms:** Attack alerts that are triggered by odd, unrecognized packets generated by certain network equipment. Load balancers often trigger it.
- **Protocol Violations:** Alerts that are caused by unrecognized network traffic often caused by poorly or oddly written client software
- **True False Positives:** Alarms that are generated by an IDS for no apparent reason. These are often caused by IDS software bugs
- **Non Malicious alarms:** Generated through some real occurrence that is non malicious in nature, possibly like our Code Red web page example above.

What are Acceptable Levels of False Alarms?

Depending on network traffic and the IDS design that is deployed, a normal IDS sensor without any customization may have only 10% of its alarms associated with a true security event. The remaining 90% of noise is not an acceptable percentage. While it may be debatable what can be considered an acceptable percentage of false alarms, with correct tuning (depending on the technology in use) an average real alarm rate of 60% or better is possible under normal conditions. I have seen real alarm rates above 90%, depending on the level of tuning and the type of traffic on a network.

False Negatives

False negative is the term used to describe a network intrusion device's inability to detect true security events under certain circumstances. In other words, malicious activity is not detected and alerted. Fortunately, there are actions that can be taken to reduce the chance of false negative conditions without increasing the number of false positives. The difficulty in creating this "balance" is to create a more manageable NIDS deployment without introducing extra risk. First, however, we need to analyze how network intrusion detection systems detect these attacks so we can understand the consequences associated with our actions.

NIDS Design Models

Network intrusion detection systems (NIDS) generally follow one of three design models. Each design model has its own strengths and weaknesses and many devices are a combination of the three models. These general IDS design categories are signature-based, anomaly-based and protocol modeling.

Signature-Based NIDS

This technology analyzes packets for specific patterns related to known attacks. This is the most common design: almost all NIDS devices have a strong dependence on signature-based detection at some level. Signature-based detection is relatively easy to understand, deploy, and update, and is good at positively identifying known attacks. However, one drawback to signature-based systems is that they may not detect unknown or modified attacks. The Code Red worm can be used as a simple example of this.

The Code Red worm initially contained a payload with the attack 'www.worm.com', so initially a signature could be written that would trigger an alert on any traffic with 'www.worm.com' in the payload. However, this attack could be changed to contain worm.net in the payload. Therefore, the signature triggering on 'www.worm.com' would be useless and would generate a false negative condition, which is to say that traffic that was an attack was not detected

The effectiveness of signature-based detection is also diminished by the fact that the NIDS may pick up the attack signature in non-attack traffic. For instance, a user could post an article on an internal web site containing Code Red information. This site could contain an analysis of the attack that contained 'www.worm.com' in the text. This would then be triggered anytime anyone accessed this web site - creating a false alarm each time. These false alarm conditions are relatively easy to introduce in signature-based NIDS devices. However, this should not deter users from employing signature-based designs, as they can create signatures that are broad enough to catch most attacks thus reducing false negatives.

Anomaly-Based NIDS

This type of NIDS design is based the type of activity that normally occurs on a network: the NIDS considers the amount and type of normal network traffic and triggers alarms on the detection of unusual activity. The usual method is by mathematically weighting the normal activity, weighting the probability of certain occurrences of other traffic, and then alerting. In this design, traffic that is unusual and occurs once or twice a day would be weighted higher than traffic that occurs fifty or a hundred times a day.

An example of this design is the SPADE (Statistical Packet Anomaly Detection Engine), a pre-processor for Snort. At first glance this appears to have an advantage over signature-based models by increasing the likelihood of catching the unknown attack, therefore reducing the potential for false negative conditions. However, to their sophisticated mathematically oriented engines, they may not be as flexible or as their signature-based counterparts.

Protocol Modeling

Protocol modeling is performed by analyzing network traffic for abnormal protocol activity and alerting on traffic with certain designated protocols or protocols that are unknown to the system. Protocol modeling relies on several different data sources to determine what is normal protocol activity. Common sources for this data may include protocol specification RFCs, popular applications that use that protocol, and thorough analysis of normal network traffic.

An example of this open interpretation is RFC 2616 for HTTP/1.1. The RFC clearly states that the http protocol places no limit on the length or the URI and that the limitation is server specific.

However, there is a note that states: "servers ought to be cautious about depending on URI lengths above 255 bytes, because of older clients." This is important when using the Code Red worm as an example because the Code Red worm overfilled a buffer with approximately 240 characters.

This combined with the normal request would put the overall URI length very close to 255 characters. In the case of the Code Red worm it may have been able to detect the buffer overflow for its overly long URL request (depending on the protocol modeling implementation) without having a signature for it.

Using the example from earlier discussion of signature-based designs, protocol modeling should not alert on the 'www.worm.com' in the html of the web page because it is within normal protocol activity.

In this situation, protocol modeling could react very well. However, there can be problems with the interpretation of the specifications.

Under these circumstances a protocol modeling-based NIDS device could be susceptible to true "false positives" in which the only fix is to wait for a vendor supplied patch.

Common Causes of False Negatives

False negatives, which are more difficult to quantify than false alarms, can be defined as an intrusion detection device not issuing alerts on legitimate attacks. There are a number of potential reasons for this, including:

- **Network design issues:** Network design flaws such as improper port spanning on switches and traffic exceeding the ability of a switch or hub contribute to these problems. Other problems include multiple entry point networks where the NIDS device cannot see all incoming and outgoing traffic.
- **Encrypted traffic design flaws:** These problems arise because the IDS is unable to understand encrypted traffic. Placing the NIDS behind VPN termination points and use of SSL accelerators are good ways to ensure the NIDS is understands all traffic.
- **Lack of change control:** Many times false negative conditions are created by a lack of communication between IS departments, networking, and security staff. Most of the time this is in the form of network or server changes that are not properly communicated to security staff. As a result, security is not able to implement measures to mitigate the risk associated with changes in security posture.
- **Improperly written signatures:** Although the attack is known and the signature is developed, the signature does not properly catch the attack or mutations of the attack because it has not been written properly.
- **Unpublicized attack:** The attack is not publicly known, therefore vendors have no knowledge and no signature is developed.
- **Poor NIDS device management:** For a variety of reasons, the NIDS device may not be properly configured. Contributing factors include:
    - Exclusionary rules to reduce false alarms that are too general;
    - The device is under too much load and cannot properly process all data;
    - Alarming is not configured properly; and,
    - The system administrator has a poor understanding of the vulnerabilities and threats associated with specific attacks.
- **NIDS design flaw:** The NIDS device simply does not catch the attack due to poor design or signature implementation.

We can appreciably reduce the risk associated with false negatives through proper device maintenance, management, design, written signatures and strong inter-departmental communication. The earlier example demonstrated how different NIDS designs are susceptible to false negatives. To reduce false negative conditions it is essential to understand the device's weaknesses and implementation issues that can reduce its effectiveness.

Kevin Timm is a Network Security Engineer at Netsolve Incorporated in Austin TX.

last updated September 11, 2001