Continuing the Automation of Network Security Process

Ken Langley
GSEC Practical v 1.4b
03/24/2004

Introduction

Managing and responding to security threats on a large, publicly accessible university network can be a daunting task. The toolsets that help provide a consistent and reliable security stance have developed significantly from basic network filtering at the router and firewall level. Network intrusion detection systems (IDS), such as Snort, provide a mechanism to discover possible attacks and automatically alert security personnel. These systems still require manual intervention to stop the threat. Once discovered, the appropriate security personnel must take corrective action. IDS technology represents the first component in automating the security process.

The next step beyond the IDS is a network intrusion prevention system (IPS) like the TippingPoint UnityOne network gear. IPS technology provides automatic security threat discovery, security alerting, and the capability to limit or block the threat at the network level. It is this automation of the complete security process from an IPS that provides a better overall defense to network threats and eases the security administrator's burden on a large open network.

Process Automation

The definition of automation includes the following statement: "automatically controlled operation of an apparatus, process, or system by mechanical or electronic devices that take the place of human organs of observation, effort, and decision." (Merriam Webster) Automation provides a more consistent, reliable process than possible with purely human effort. Therefore security process automation is using systems that replace the human effort in terms of threat discovery and response. Administrators focus on developing the security stance of an organization and using tools, such as IDS and IPS, to implement the required notification and control mechanisms.

An IDS deployment represents an initial step for automation. Using our earlier definition, an IDS sensor performs observation and decision processes based on its configuration. The sensor captures network data (observation), applies rules to determine if the packet represents a threat (decision), and then passes on the effort component to security staff via logging and alerting.

The area not addressed by IDS is the effort component. An IPS system performs the observation, decision, and effort. It responds to network threats by a combination of rate-limiting network traffic, blocking data at the switch level, or tearing down the connection.

This paper considers three security process models within the context of an open network at a large university. The three models are: manual process, or no automation support; Snort-based IDS sensors, chosen due to its popularity in a campus environment; and the TippingPoint UnityOne IPS appliance. A university network is used for a network model because it has unique properties and problems that increase the workload required by security administrators and are not easily solvable by brute force solutions.

University Network

A large university or college network is difficult for security personnel to manage threats effectively. Due to the openness inherent in the research and educational missions of these institutions, a university may not want to tightly control its network. Few colleges and universities install firewalls at the main entrance points to their network as a result of this philosophy.

Another factor is a large resident population. The access controls present in a business or employee setting may not be appropriate for students housed on a campus environment. Students view living in a dormitory setting just like being at home and therefore expect greater latitude in network access, especially if the college or university requires them to purchase a computer. A large university may have 20,000 to 30,000 students and 5,000 to 7,000 staff and faculty members; thus the number of computing nodes active on a network can be great.

Universities and colleges may have high-speed networks to support ongoing research projects. Many large institutions are connected to both the commodity Internet as well as Internet2. The bandwidth available on a university network, combined with the large numbers of computers and the inherent open network access, can keep a security administrator extremely busy.

Example University Network

The University of North Carolina at Chapel Hill is a good example of a large university network. The network does not have a firewall at the entry point to the network. The campus has 26,000 students (UNC-CH *Quick Facts*), most who have a personal computer as a part of the Carolina Computing Initiative (UNC-CH *CCI*), and 10,000 supporting faculty and staff members (UNC-CH *Information*). The connection to Internet and Internet2 is via an OC-48 (2.5 gigabit per second) interface. Buildings are linked with either switched 100 Fast Ethernet or Gigabit Ethernet (Hawkins). The fact that the overall network architecture diagram is freely available over the general network shows the openness that a large university has. This type of network, combining a large set of computers, good bandwidth, and little access control, provides a challenging environment for the security professional.

Manual Process

Intrusion detection and prevention systems exist to fill an important need for security personnel. Security staff had few tools to recognize an active security threat without an IDS or IPS in the past. A threat would be found after a human noticed it. For example, a computer might malfunction or the data network would become unavailable. The administrator would have little data to investigate the possible cause without IPS capability. Few networks had sufficient logs of past activity to diagnose an attack that had ceased. A current attack could be determined after a human read output from a packet analyzer. Very little automation was available to the network security staff.

The process to defend against an attack provided little automation as well. Administrators would develop, test, and deploy filters on network equipment such as routers and firewalls. These filters are coarse grained in nature, blocking traffic from a set of source IP addresses and ports. Polymorphic threats easily evade such a response by just changing the source port. Human error in deploying filters could cause a significant interruption in network availability. When the SQL Slammer worm struck, open network could filter out traffic destined to the suspect ports manually using router access control rules (Vamosi). However, legitimate access to services over those same ports was also stopped.

The staffing needs to handle a large university with an open network without automation tools would be large. The security staff would constantly have to monitor packet loggers looking for threat signatures. Network staff would change access control rules on critical networking equipment regularly, thereby possibly creating new problems within the network. The need to improve the security process drives the adoption of automation technologies by security administrators.

Snort

Snort is a lightweight, cross-platform NIDS based on open source technology. (Roesch) Snort is typically installed on a UNIX platform, such as GNU/Linux or BSD derivative, but can be installed on a Windows 2000 system. The software leverages several open source technologies. It uses `libpcap` for packet capture. Event logs can be stored in a relational database system, such as MySQL or Postgres. Using a database system allows the administrator to track multiple Snort sensor logs centrally. The Analysis Console for Intrusion Databases (ACID) is an open source tool to review and manage these logs. (Danilyw) ACID requires a web server configured with PHP support and ADODB. A recipe for a complete open source IDS system is available from SANS (Vanderpoel) or linked from the Snort documentation web site (Harper).

Internally, Snort consists of three subsystems on top of the libpcap packet sniffer: a packet decoder, detection engine, and the alert and logging module. The Snort packet decoder looks into the data segment of the packet. The sensor applies the rule sets in the detection engine to determine if the packet matches a signature. If a match is found an alert is sent to the administrator. The alert system supports the following alerting types:

- SMTP mail message
- Syslog
- SMB messaging using a samba client
- Custom log file

Perhaps due to its open source roots, Snort enjoys a large community supporting its ongoing development. Both the source code and its rules are open to community contribution. Currently, the rule set contains over 2200 rules. The open source community and a commercial company named Sourcefire support the underlying Snort software.

Several factors must be considered when deploying one or more Snort sensors. These factors include the placement of the sensors in the network, the performance of the equipment in evaluating the traffic it sees, and the effectiveness of the rule set implemented. Security administrators must balance these considerations carefully.

Where sensors exist on a network is crucial because that will determine the extent of data it will process. Intuitively, a sensor should be installed at major ingress and egress points into the network to see the variety of security threats. In a University environment, these points would include connections to the commodity Internet, Internet2, and large residential areas, such as campus dormitories. Answer 2.5 of the Snort FAQ addresses placement of sensors within the network and weighs the benefits and drawbacks of various configurations (Snort Core Team). It concludes with “just pick a spot you're likely to look at the logs for :-).” The use of port spanning on a switched network allows the sensor to be installed where physically convenient but still see the network traffic desired.

Performance of a sensor contributes to its effectiveness. An overloaded sensor will not be able to examine packets fast enough, drop untested traffic, and miss possible attacks. The study Characterizing the Performance of Network Intrusion Detection Sensors found that a Snort IDS sensor using version 1.9 could be overwhelmed by network traffic on a 100 megabit network. Important factors governing performance include: the number of rules, the bandwidth involved, and the hardware capacity. Large research universities have high bandwidth pipes and a large population (e.g. students), which could overload a Snort IDS, especially during a wide spread security event.

Another key performance contributor is the type of logging used. Storing event logs in a centralized database can have a significant impact. Administrators use a database to provide a single point for the event logs to reduce the burden of investigating network threats. A study in 2003 found that optimizing the database access of a sensor decreases the processing overhead by 25 percent. (Schaelike November 2003)

The detection rules activated on a sensor are important in determining how well a Snort deployment works. If a detection rule is too narrow it may miss a polymorphic attack. If a rule is too general the security staff may be inundated with false positives. The worst outcome is if a rule is configured that inadvertently allows a security threat to pass without notice, known as a false negative. Many of the rules in the Snort database provide information on possible false positives and negatives (Sourcefire), but the administrator must review and tune the rules on each sensor based on experience.

TippingPoint

The TippingPoint UnityOne IPS appliance is installed similar to network switchgear. The chassis provides front-accessible ports in a 2U to 4U rack mountable chassis. The IPS hardware is a mix of network processors, custom ASICs, and customized Linux kernel. The underlying technologies are proprietary to TippingPoint. Management of these appliances is done through a command line interface on the appliance, built in web management using the Local Security Manager software, or by an Enterprise Security Management System (SMS). The SMS system is accessed by a windows client and requires and additional network chassis to be installed, which is the SMS server. The appliances provide administrative access through a 10/100 Ethernet port, a serial interface, or a front mounted six-line LCD screen.

UnityOne appliances support a range of capacities and network interface technologies. The entry level 200 model supports four 10/100 Ethernet using copper connections (RJ-45) with an aggregate throughput of 200 megabits per second. The enterprise level 2400 model has eight 10/100/1000 ports using either copper or fiber connections with an aggregate throughput of 2.0 gigabits per second. The UnityOne appliance requires that ports be used in pairs. One port is inbound connections and one for outbound. If the appliance is to be used in a passive mode (e.g. connected to a SPAN port on a switch) the remaining port of the pair is unused.

The IPS contains dual power supplies to increase its reliability and has an option named Zero Power HA to allow traffic to flow in case of complete shutdown. They support clustering in either active-passive or active-active mode. If an internal failure occurs the appliances are configured to “fail open” and allow network traffic through without applying security filters (TippingPoint *UnityOne Appliance Datasheet*). Clearly, these devices are directed at networks with high availability and reliability needs.

The UnityOne IPS provides the same capabilities as the aforementioned Snort IDS. The appliance provides packet capturing and decoding, attack detection, plus logging and alerting. It extends the feature set of traditional IDS sensors by developing a network engine that can limit or block network traffic. The UnityOne IPS consists of three major systems: the Management Processor (MP), which provides the software interface to control and configure the device; the Threat Suppression Engine (TSE), which provides the high speed engine for threat detection; and the Multi-Zone Defense Module (MZD), which acts as a layer 2 network switch and connects the unit to the network (NSS Group). The TSE acts similar to the rule detection subsystem in a Snort IDS. The MZD acts as the network gatekeeper responding as directed by the TSE.

The IPS appliance can generate the standard alert and log events that a Snort IDS provides, but it provides other features beyond a typical IDS. The IPS can also block network data, rate-limit traffic, or generate a packet trace log with all or part of suspicious packets for later analysis. If an alert is needed, the administrator can choose from the following alert systems:

- SMTP mail message
- Pager notification
- SNMP event
- Syslog
- Custom scripting

The same considerations of placement, performance, and effective detection rules are important in IPS deployment. The UnityOne appliance is designed to be installed into the network data stream as an inline component and distributed throughout the network. Network segments beyond just ingress and egress points are good candidates for where the IPS should be placed. The IPS can be installed inline to enable traffic blocking or in a passive mode on the network. Passive installations allow the administrator to gain confidence in the equipment before activating it blocking capabilities. Bandwidth concerns are addressed by sizing the appliance appropriately to the network segment.

A security administrator can use one or more of the IPS appliances to create different security zones within the network. These zones can implement different security requirements to better match the needs of network users. In a university deployment, security personnel can implement controls appropriate to residential student use that might different from that of employee activity, and still maintain the overall openness of the network.

The performance of the UnityOne IPS is still a consideration for installation. The rated aggregate throughput of the 2400 model is 2.0 gigabits per second across all network ports and rated maximum latency of 215 microseconds. A recent review (NSS Group) found that the UnityOne 1200 performed well up to the maximum test input of 1 gigabit per second with no degradation in packet filtering. Latency under testing did not increase beyond 116 microseconds during a “half-load” test with a packet size of 1514 bytes.

How the IPS appliance deals with large numbers of connections is another concern. However, testing shows that the UnityOne 1200 can manage 1,000,000 simultaneous connections and still block threats without hindering legitimate traffic (NSS Group). This capacity is needed on large networks with thousands of hosts.

Developing effective rules to manage security threats on the UnityOne IPS is similar to rule development on a Snort IDS. The security administrator must review rules to make sure they are general enough to manage attacks, reduce the false positive rate, and mitigate the chance of false negatives.

The blocking action provides a new area of concern. A poorly implemented blocking rule can have a traumatic affect on the network. A blocking rule that is not general enough does not provide the protection that the security personnel expect. A rule with false positives may cause the IPS to perform a denial of service (DoS) attack on a legitimate network service.

Worms

Why adopt a new technology such as a TippingPoint IPS? Administrators must see a clear benefit to installing new equipment. If an IDS sensor provided enough features to handle security threats, overloaded security staff would not use a new device. A fast-moving widespread threat that proved largely intractable to handle without an IPS would demonstrate its need.

Compelling reasons for installing such an appliance arrived in the Fall of 2003. The Blaster worm was released on August 11, 2003 and it devastated unprotected networks as infected machines flooded local subnet spreading the infection and attempted a DoS attack (Knowles). Security administrators supporting unprotected networks were largely unable to check its initial spread due to its fast infection rate. Snort signatures were developed quickly that could detect the Blaster payload, but the damage was done and the next task was to develop control mechanisms to limit further damage until the virus could be removed. Those universities that already had a TippingPoint appliance were able to escape relatively unharmed (TippingPoint *Six New Customers*). Many universities began investigating the TippingPoint appliances shortly after.

On August 18, 2003 the next large scale threat was released, named Sobig.F by anti-virus companies (Nahorney). TippingPoint appliances again proved their value during this time. John L. Oberlin, associate vice chancellor for IT at University of North Carolina at Chapel Hill, stated “The UnityOne was so effective at blocking the virus that we immediately purchased several appliances in order to protect the entire network (TippingPoint *Six New Customers*).”

Operational Comparison

The UnityOne appliance and Snort system have significant operational differences in the areas of device management, security, and filter maintenance.

A typical Snort deployment includes one or more sensors, an ACID console, and a database to store events. Several software packages must now be managed to keep the IDS system current. A sensor requires libpcap and Snort to do its work. The console needs the ACID package that is built out of PHP, web server, PHPlot, ADODB, JGraph, and GD, and a database system. All of these packages are provided by different projects. Many open source projects experience rapid change and keeping the software up to date can be challenging. The large number of separate components reflects the nature of open source projects. These projects have a desire to leverage work already done by others to speed up implementation and to encourage administrators to mix and match tools that better fit their environment.

In contrast, the UnityOne IPS appliance provides a single, comprehensive interface to all of its packages. All of its software is updated by one process and is only available from TippingPoint. The toolset provided by the appliance is the only one the administrator can use. A commercial package is expected to deliver decreased dependencies and provide a more integrated approach to systems management.

A Snort administrator manages three different password realms: the operating system of the sensor, the web server controlling access to ACID, and the database system storing logs. Each component has different mechanisms to implement and manage passwords. The TippingPoint appliance utilizes one password realm. It provides fine-grained access control with roles and the ability to restrict users to specific IPS appliances and network segments within an appliance.

Implementing new detection rules is critical in running both IDS and IPS systems. The utility of such a system decreases if attack signatures are not kept current. The Snort sensor rules are contained in a text file named `snort.conf`. New rules are implemented by updating this file and then restarting the snort process (Snort Core Team). The Snort web site tracks the current set of default rules and makes them available for download via HTTP or CVS (Snort Core Team). Administrators can add custom rules using the Snort rules description language. (Roesch and Green)

TippingPoint provides a service called the Threat Management Center to develop signatures for administrators on a subscription basis. The administrator can configure the devices to automatically update the threat signature database, known as Digital Vaccine, when a new version is available or download and install them manually (TippingPoint *Digital Vaccine*). Local signatures are developed using the Custom Shield Writer program.

A Snort system allows the administrator wide latitude in terms of deployment decisions (e.g. which database system to use, which operating system will sensors have) but at the cost of increased management requirements. The UnityOne appliance provides a different approach by limiting flexibility but also providing a more unified management experience. Much of this difference can be attributed to the cultures in which these systems are developed and deployed. The administrator must consider the amount of work required to manage these systems when installing within his environment.

Conclusion

An IPS device, similar to the TippingPoint UnityOne appliance, provides important new features beyond those of a traditional IDS. The added ability to have the threat detection equipment rate-limit or stop network traffic allows the security staff to automate more fully its responses to threats. A UnityOne IPS appliance enables the staff to focus on developing a good security posture and implementing it one time, in the IPS. The appliance performs the enforcement of these decisions automatically, consistently, and continuously.

Care must be taken when using an IPS. A poorly implemented or badly configured rule could have an extremely negative impact on legitimate use of the network. However, a security team that uses a well-implemented IPS can provide greater control over the internal network segments and not just the ingress and egress points typically protected by firewalls. This technology clearly reduces the workload of the security staff and the risk profile of the protected network.

References

Danilyw, Roman. “Analysis Console for Intrusion Databases”. URL: http://www.cert.org/kb/acid/ (3 March 2004).

Harper, Patrick S. “Snort, Apache, PHP, MySQL, ACID on Redhat 9.0 Installation Guide”. 6 October 2003. URL: http://www.snort.org/docs/snort_acid_rh9.pdf (22 March 2004).

Hawkins, Mike. “Data Network Diagram.” January 2003. URL: http://www.unc.edu/depts/oit/ns/gifs/maincampus-2003.gif (22 March 2004).

Knowles, Douglas. Frederic Perriot, and Peter Szor. “Symantec Security Response - W32.Blaster.Worm”. 26 February 2004. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html (22 March 2004).

Merriam-Webster OnLine. “Automation”. URL: http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=automation (20 March 2004).

Nahorney, Benjamin and Atli Gudmundsson. “Symantec Security Response – W32.Sobig.F@mm.” 8 January 2004. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html (22 March 2004).

NSS Group. “TippingPoint UnityOne-1200 V1.4 Technical Evaluation “. URL: http://tippingpoint.com/resource_library/pdfs/NSSUnityOneWP.pdf (20 March 2004).

Roesch, Martin. “Snort - Lightweight Intrusion Detection for Networks”. URL: http://www.snort.org/docs/lisapaper.txt (3 March 2004).

Roesch, Martin, and Chris Green. “Snort Users Manual 2.1.1.” URL: http://www.snort.org/docs/snort_manual/ (22 March 2004).

Schaelike, Lambert, Thomas Slabach, Branden Moore, and Curt Freeland. “Characterizing the Performance of Network Intrusion Detection Sensors.” July

2003. URL: http://www.cse.nd.edu/~spanids/papers/nids_perf_raid03.pdf (15 March 2004).

Schaelike, Lambert, Matthew R. Geiger, Curt J. Freeland. “Improving the Database Logging Performance of the Snort Network Intrusion Detection Sensor”. November 2003. URL: http://www.cse.nd.edu/~lambert/pdf/TR-0310.pdf (20 March 2004).

Sourcefire Research Team, “Snort Signature P2P GET”. URL: http://www.snort.org/snort-db/sid.html?sid=1432 (20 March 2004).

Snort Core Team. “The Snort FAQ.” 9 April 2003. URL: http://www.snort.org/docs/FAQ.txt (15 March 2004).

TippingPoint Inc. “Tipping Point FAQ.” URL: http://tippingpoint.com/resource_library/pdfs/TPTI-FAQS.pdf (21 March 2004).

TippingPoint Inc. “TippingPoint Technology Protects Six New University Customers.” 4 September 2003. URL: http://www.tippingpoint.com/news_events/pdf/UnivCust_090403.pdf (20 March 2004).

TippingPoint Inc. “Tipping Point UnityOne Appliances Data Sheet.” URL: http://tippingpoint.com/resource_library/pdfs/200-400-1200-2400_DataSheet.pdf (20 March 2004).

TippingPoint Inc. “TippingPoint Digital Vaccine.” URL: http://tippingpoint.com/products/unityone_digital_vaccine.html (20 March 2004).

University of North Carolina at Chapel Hill. “Quick Facts about Carolina.” 27 June 2003. URL: http://www.unc.edu/depts/design/quickfacts/ (20 March 2004).

University of North Carolina at Chapel Hill. “Information about the University.” 1 December 2003. URL: http://hr.unc.edu/jobseekers/info-univ.htm (20 March 2004).

University of North Carolina at Chapel Hill. “Carolina Computing Initiative.” URL: http://www.unc.edu/cci (15 March 2004).

Vamosi, Robert. “SQL Slammer: How it works--prevent it.” 27 January 2003. URL: http://zdnet.com.com/2100-1105-982226.html (22 March 2004).

Vanderpoel, TJ. “Deploying Open Sourced Network Intrusion Detection for the Enterprise.” 4 March 2001. URL: http://www.sans.org/resources/idfaq/open_source.php (20 March 2004).