

Building a High-Performance Unified Threat Management Appliance

Part II: Software Issues

by Mick Johnson, Sensory Networks

Multiple high-speed applications requires a dedicated hardware co-processor

Keeping your network fast and secure is crucial to the success of your business. Who hasn't had their day ruined because the network was down, slow, infected or in the midst of repair? While the twin needs of speed and security are clearly important to all of us, the difficulty of providing both at the same time often escapes our attention. The security appliance market is experiencing three main drivers right now: devices must go faster, secure the network against more threats, and contain more functions. These integrated security appliances which target all three at once are commonly called unified threat management (UTM) appliances, and have the highest growth rate in the security appliance market according to research groups such as In-Stat¹ and IDC². However, satisfying the three market drivers is a complex task for appliance developers, with several key hardware and software issues to be solved.

The processing of network traffic can be divided into 2 core sub-tasks: *collection* and *detection*. Collection refers to capturing traffic from the wire, parsing packet headers, processing the network stack, discerning which sequence of bits comprise packet payloads, re-injecting packets, and so on. Detection is the problem of taking those payloads and determining whether the packet or stream is malignant or benign. Detection represents the greatest workload for today's devices, and in particular UTM appliances. This can clearly be seen in recent tests³, run by Network Computing, of the market-leading firewall appliances, where performance fell dramatically as soon as deep-packet inspection was enabled. The work required in the detection phase had a massive impact on performance on all of these devices.

The difficulty in scanning network streams at wire speeds for malicious data arises from limits on host memory bandwidth and latency. Typical approaches to network security incorporate signatures for detection of malware signatures targeting some vulnerability in the target software, protocol or system. As the number of these signatures grows it becomes impossible to fit any but the most rudimentary databases into level-1 and level-2 cache, meaning scanning each packet requires reads from main memory, or even disk! This I/O bottleneck severely limits throughput once large commercial signature databases are loaded. Jitter in processing time is particularly apparent in software-only systems, where it has a knock-on effect on TCP throughput due to the latter's exponential back-off. A longer processing time on a particular packet in the stream doubles the window size and fails to collapse it effectively for future packets with faster processing. High-performance UTM appliances need bounded latency for content scanning, and the capability to match against multiple massive signature databases. For more information on the hardware issues, please refer to part I. <http://www.analogzone.com/nett0529.pdf>

It is insufficient to merely bundle multiple applications on a single device, even one with a dedicated security co-processor, and expect them all to run at gigabit speeds. There are two primary software issues that must be dealt with; the first is the integration of each security application into a co-processing platform in such a way that the level of coverage is maintained while performance is increased; the second is the efficient combination of multiple security applications, which need to scan the same traffic in different ways to discern different threats.

The software engine must see only malicious traffic

The vast majority of traffic that passes through a business or enterprise network over the course of a year is clean. Hardly surprising when we're all paying for bandwidth! Web is still the dominant class of traffic on the vast majority of backbone links, although this is starting to be challenged by P2P. Over 90% of backbone traffic is TCP⁴. The key task, therefore, is accelerating the scanning of clean traffic while still detecting malicious traffic when it does arrive. Such an approach delivers on the two key ideals of a UTM device: performance and coverage.

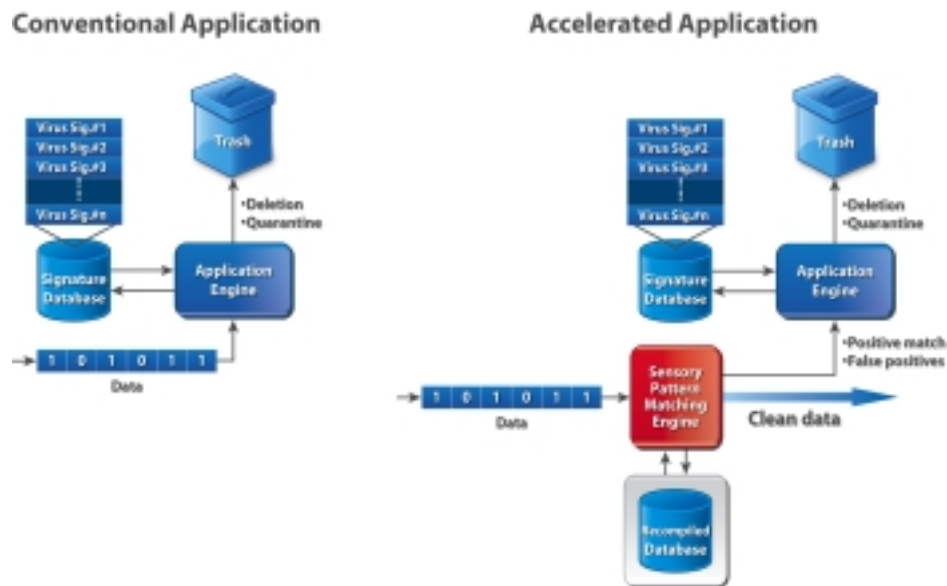


Fig. 1: Accelerating Security Applications

Most security applications inspect network traffic data in several ways, including pattern-matching against a signature database, analyzing various statistical characteristics, and executing programs within a protected operating environment. If all these techniques were to be applied to every byte of traffic then multi-gigabit speed would be impossible. With a specialized co-processor capable of scanning content at multi-gigabit speeds an effective technique is to filter first on signatures, and then pass only those traffic streams regarded as suspicious to a more sophisticated but slower application engine (see Fig. 1). Under this scheme the speed-up of the system of a whole depends on three things: the speed of the accelerated engine, the speed of the normal engine, and the proportion of traffic that bypasses the slower part.

Amdahl's law⁵ shows us that the proportion of traffic that bypasses the slower application engine is of major importance. At 100% bypass, throughput is equivalent to the hardware engine alone, but as this proportion drops so too does the speed-up. Traffic regarded as suspicious by the hardware, but which is actually benign, can be termed a *false positive* within this system, and the first key software issue is to minimize these false positives.

The rate of false positives depends primarily on the disparity between the software and hardware databases. The hardware database must define a superset of malicious traffic lest some malicious data evade the system. The false positives come from the set of traffic matching only the hardware database and, as this set increases in size, the speed-up falls. However, this drop is also proportional to the amount of traffic matching false positive; a single signature may match a high proportion of traffic, while the other signatures match very little. In the worst case, a single signature might match against all network traffic passing through the system! An example of such a signature can be found in the deleted.rules set of the Snort database.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS
header field buffer overflow attempt"; flow:to_server,established;
content:"|3A|"; content:"|0A|"; content:"|00|";
reference:bugtraq,4476; reference:cve,2002-0150;classtype:web-
application-attack; sid:1768; rev:7;)
```

This signature matches any packet going to a Web-server that contains a colon, a newline character, and a NULL – every single packet.. A single bad signature can have a ruinous effect on an accelerated system. It is therefore crucial to maintain a high degree of quality control in terms of signature deployment.

Each layer must guarantee security and reduce workload

To deliver on the promise of reducing the number of devices actually deployed, a UTM appliance must include at least three things: IPS/IDS, Antivirus scanning of HTTP and SMTP traffic, and antispam-filtering of e-mail. It might also include features such as quality-of-service firewall/VPN functionality or antispyware scanning. These applications must all inspect traffic running over various protocols (see Fig. 2) but often need to transform the data before the inspection can proceed.

	IPS/IDS	Antivirus	Anti spam
HTTP	Yes	Yes	No
SMTP	Yes	Yes	Yes
Other protocols	Yes	No	No

Fig. 2: Protocol Vs Application

For example, a single SMTP conversation could exploit buffer-overflow vulnerabilities in the SMTP server, contain a Trojan in an attachment, or simply be spam. To be effective a UTM device must scan for all three types of threat in all SMTP conversations. The first

threat can be discovered with rules such as the following, from the Snort IPS engine's smtp.rules⁶ file, which detects attacks targeting older version of sendmail. To detect such attacks, this signature must be applied to TCP packets and reassembled TCP streams.

```
alert tcp $EXTERNAL_NET any -> $SMTP_SERVERS 25 (msg:"SMTP sendmail
5.6.5 exploit"; flow:to_server,established;
content:"MAIL FROM|3A| |7C|/usr/ucb/tail"; nocase;
reference:arachnids,122; classtype:attempted-user; sid:665; rev:5;)
```

Once the SMTP conversation has been verified as safe, the next step is to determine whether the e-mail is valid or not. This is done using a wide range of techniques, but they all typically require that the message be reassembled, whereupon fields such as the message-headers are inspected. The following is a standard rule from the 20_drugs.cf⁷ database in the popular open-source Apache SpamAssassin project, designed to detect Viagra spam.

```
body ONLINE_PHARMACY
/\bonline pharmacy|\b(?:drugs|medications) online/i
```

In addition to antispam processing, e-mail attachments must be scanned for the presence of viruses or Trojans. The following is a virus signature from the ClamAV database⁸, designed to detect the downloader of a Win32 dialer from a malicious website. The recursive nature of MIME encoding requires that the message be parsed and the attachment decoded before it can be inspected.

```
Trojan.Downloader.Small-240
(Clam)=323cb9fcdcc2030b72337633f634a0a032bafa34fd5f4001d6685e23407f014e
255c777e646d2e65786556ffb7ff00687474703a2f2f730b2d7069652e636f6d2f74be6
50bd0ea662fa0302f2818c091bf3f202d313c1353906e90db0553c00b1d0b0406690664
968d088e400664408f9064400664
```

For a given e-mail to pass through a UTM appliance it must pass all three of the checks described above.

The second key software issue is to minimize workload and maximize throughput without missing attacks. This must take into account where each application hooks into the network stack, what transformations it applies to the data, and what security guarantees it provides to later levels.

In the case of e-mail, the IPS engine alerts and drops TCP connections that would overrun the remote server before the message can be scanned. Only after the connection is assured to be safe does the next layer of scanning begin. At each subsequent stage, less malignant traffic passes a given application, increasing speed and reducing risk. Appliance vendors wanting to store junk e-mail will also need to decode attachments and scan for viruses, regardless of whether that message was classified as spam or not. The key to solving this issue is having an integrated, extensible architecture that can plug-in all the necessary components. By using such a framework, appliance developers can

rapidly include their desired security functions and have them work in combination, without sacrificing either speed or coverage.

Developers of high-performance UTM appliances are turning to co-processing platforms to deliver the necessary functions at multi-gigabit speeds. Using these platforms effectively requires that each security application running on the device be integrated both with the hardware acceleration libraries and each other. Acceleration is severely inhibited by false positives between the hardware and software databases, which in turn derive from poor signature quality. A layered approach is required to combine applications effectively, where each layer provides security guarantees and removes traffic inapplicable to subsequent layers.

About the Author

Mick Johnson is the Product Marketing Manager for Sensory Networks, the leading OEM provider of high performance network security acceleration technology. The company's NodalCore hardware acceleration products include a broad range of chipsets, accelerated software libraries, PCI acceleration cards and appliance platforms for Antivirus, Antispam, Antispyware, Content Filtering, Firewalls and Intrusion Detection/Prevention systems. He has a BSc University Medal in Computer Science from the University of Sydney, and can be reached at mick@sensorynetworks.com.

References

¹ Victoria Fodale, Integrated Security Appliances: SMBs Fuel Explosive Growth

² Charles J. Kolodgy, Worldwide Threat Management Security Appliances 2005-2009 Forecast and 2004 Vendor Shares

³ Adrian Peters & Michael Jones, Network Computing, Deep Inspection Firewalls: Clash of the Titans, 2005

⁴ Packet Level Traffic Measurements from the Sprint IP Backbone – Fraleigh, Moon, Lyles, et al. Sprint Labs, 2003

⁵ Gene Amdahl, Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities, AFIPS Conference Proceedings, (30), pp. 483-485, 1967

⁶ A description of this rule can be found at <http://snort.org/pub-bin/signs.cgi?sid=655>

⁷ This ruleset can be found at http://spamassassin.apache.org/full/3.0.x/dist/rules/20_drugs.cf

SpamAssassin is a trademark of the Apache Software Foundation

⁸ A description of this virus can be found at <http://www.viruslist.com/en/viruses/encyclopedia?virusid=68781>

