The image contains a list of topics covered in the Network Box technical review for 2013. Here is the text in a readable format:

- 2013 Threat Round-up
- DNS, Mail & 512 bytes
- Secure Socket Layer Security Strategy (SSL Plus)
- Java Vulnerabilities and Exploits
- How to be a Prepper (aka How to survive a DDoS Attack)
- Real-Time Detection and Blocking of Outbound Trojan Activity
- OWASP Top 10 2013
- NBRS-5.0 Application Identification Framework
- NBRS-5.0 Mail Scanning
- IDS/IPS vs WAF
- Network Box 5 Software Platform
- Network Box 5 Hardware Platform
- Network Box 5 SSL Proxy
- Focus on the Next Attack (not just the last one)
2013 Threat Round-Up

Summary and analysis of the Network Box Threat Statistics for 2013

<table>
<thead>
<tr>
<th>Network Box Threat Statistics</th>
<th>2012 Numbers</th>
<th>2013 Numbers</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUSH Updates</td>
<td>6,328</td>
<td>5,995</td>
<td>-5.9</td>
</tr>
<tr>
<td>Signatures Released</td>
<td>4,484,811</td>
<td>6,860,044</td>
<td>+53.0</td>
</tr>
<tr>
<td>Firewall Blocks (/box)</td>
<td>10,497,946</td>
<td>10,544,863</td>
<td>+0.4</td>
</tr>
<tr>
<td>IDP Blocks (/box)</td>
<td>1,669,242</td>
<td>1,227,740</td>
<td>-26.4</td>
</tr>
<tr>
<td>Spams (/box)</td>
<td>163,126</td>
<td>172,604</td>
<td>+5.8</td>
</tr>
<tr>
<td>Malware (/box)</td>
<td>7,470</td>
<td>19,793</td>
<td>+165.0</td>
</tr>
<tr>
<td>URL Blocks (/box)</td>
<td>1,989,761</td>
<td>1,858,598</td>
<td>-6.6</td>
</tr>
<tr>
<td>URL Visits (/box)</td>
<td>50,247,987</td>
<td>39,448,903</td>
<td>-21.5</td>
</tr>
</tbody>
</table>

PUSH Updates & Signatures Released
During 2013, Network Box Security Response PUSHed out 5,995 updates, totaling 6,860,044 signatures (down 5.9%, and up 53.0% respectively, compared with 2012).

Firewall & IDP Blocks
During 2013, the average Network Box blocked 10,544,863 attacks using firewall technology, and 1,227,740 attacks using IDP technology (up 0.4% and down 26.4% respectively, compared with 2012).

Spam & Malware
During 2013, the average Network Box blocked 172,604 spams and 19,793 malwares (up 5.8% and 165.0% respectively, compared with 2012).

URL Blocks & URL Visits
During 2013, the average Network Box blocked 1,858,598 websites due to company content filtering policy enforcement, with 39,448,903 website URLs visited over the year (down 6.6% and 21.5% respectively, compared with 2012).

As always, every month we see more and more threats, with faster distribution times. Network Box will continue to invest in technologies to speed-up the protection release cycle, and will continue to leverage our excellent customer relationships so that we can all work together to co-ordinate an effective defense.
DNS solves the core problem of associating a name with an IP address, but also goes beyond that by allowing the name to also retrieve other informational records (such as MX records for mail servers). Various conventions have arisen, over time, to extend the record types understood by DNS clients, and how they are interpreted. Nowadays, DNS is ubiquitous. It is the ‘telephone directory’ of the global Internet.

But, being designed back in 1982 for a very limited number of records, DNS has a core foundational problem – and that is it is built primarily on the UDP/IP protocol, which means responses must typically be restricted to 512 bytes or less. DNS clients can fall-back to TCP/IP to overcome this restriction, but that is comparatively very slow and problematic for some deployments (not to mention not universally enabled, and often firewall blocked). When DNS responses just consisted of a few short IP address records, this wasn’t a problem. Now, with MX records, A records, AAAA records, NS records, and most importantly DNSSEC, the responses are exceeding 512 bytes, and DNS is struggling.

Historically, it has been relatively common to permit UDP/53 (DNS over UDP) at the firewall level, but deny TCP/53 (DNS over TCP) as the TCP protocol was used for larger DNS zone transfers. Denying TCP/53 offered a simple approach to firewalling sensitive DNS zone transfers. Lately, some major DNS providers are going even further by restricting certain DNS query types (such as the “ANY” query used to ask a server for all records it has on a particular name). Both of these defensive approaches are violating the RFCs, and resulting in undeliverable eMail and general network connectivity problems.

Network Box has historically tried to balance the relatively high latency of the DNS system with efficiency, in order to enhance performance and optimize network usage. It is, quite simply, more efficient to send a single request for multiple record types in a single response, than to request each record type individually. Specifically, we have used DNS “ANY” queries when looking up mail server MX records (as it allows us to retrieve all relevant records with a single query). This approach, while fully standards-comformant, is becoming problematic when dealing with badly configured name servers, and their firewall protections, that don't conform to the Internet standards. Network Box often receives the blame for such situations, even though it is clearly the remote DNS server that is at fault. The situation is only likely to get worse, as more and more sites adopt DNSSEC and DNS replies get larger.

With the February 2013 Patch Tuesday, Network Box is releasing a revised version of our SMTP store-and-forward server that no longer uses “ANY” style DNS queries. Instead, it will issue multiple individual record type queries, as needed. This will reduce the average response size of DNS packets (as only the record types specifically asked for will be returned), but with the drawback that more DNS queries (with associated slow latency) may be required to retrieve other record types. This approach is fully standards-compliant, and improves compatibility with existing non-compliant DNS servers – but at the expense of more DNS queries for some types of DNS server responses.

The Network Box Transparent SMTP system is not affected by this change, as it does not use DNS (instead relying on the DNS resolution of the host making the original TCP/53 SMTP mail service connection).

As always, should you have any questions or concerns, please feel free to contact your local Network Box NOC for assistance.
Network Box
Secure Socket Layer Security Strategy
(SSL Plus)

This is the first article in a series about the Network Box Secure Socket Layer (SSL) Security Strategy, or SSL Plus. SSL Plus encompasses certain technologies present in Network Box NBRS-5.0 and a set of business processes employed by Network Box Corporation, that add value to the essential security provided by the SSL.

This first article in the series was intended to be an introduction to some of the most important technologies falling under the umbrella of the term Secure Sockets Layer and SSL Plus, but in recent weeks an incident occurred that not only highlights the delicate nature of SSL but also the material benefit that Network Box NBRS-5.0 can bring to customers.

No time will be spent on definitions and explanations of the technologies mentioned in this article.

The incident in question is the TurkTrust Certificate Authority’s accidental issuing of an Intermediate Certificate to a client instead of a regular Website Certificate. The details of this incident, including the causes and the presence of any malice are not important, but what is important is the outcome that the runaway Intermediate Certificate was able to sign any Website Certificate and that browsers accepted these Website Certificates as perfectly valid. This incident once again demonstrates the weakness of Intermediate Certificates and indeed Certificate Authority as a whole, as well as the fact that Web Browsers place so much trust in Certificates until that trust is explicitly and ponderously removed.

The incident hit mass media at the beginning of January and it is only a month later, that major Operating System vendors are releasing updates to remove trust in these Intermediate Certificates. Although some of the major Web Browser vendors were quicker to respond, the issue still remains that organizations and end users will not be anywhere near as quick to globally update their own software.

Let us now examine the case where an organization is using the Network Box NBRS-5.0 Web Client Scan product, that includes Client SSL Plus. One of the features of Client SSL Plus is it’s ability to inspect SSL connections on behalf of all Web Browsers in an organization, and apply administrator defined Access Control Rules to those connections at an organizational level, at the Network Gateway. If any secure connection is identified as having the runaway Intermediate Certificate in its chain of trust, then the Access Control Rule can have the connection blocked by the NBRS-5.0 Network Box. Because the Intermediate Certificate is identified using a globally unique SHA1 fingerprint, the Access Control Rule will have no affect on any other secure connections.

Upon learning of this incident, Network Box Regional SOC staff will communicate with a customer and approve the Access Control Rule rule for immediate activation on the customers NBRS-5.0 installation, thus protecting that ONE Network Box customer in a matter of MINUTES.

Upon learning of this incident, Network Box Headquarters will initiate a procedure to install the Access Control Rule as a global default setting and using the NBRS-5.0 Configuration Synchronization, propogating that protective rule to all CURRENT Network Box customers in a matter of A FEW HOURS.

Upon learning of this incident, the Network Box Core Development Team will integrate the Access Control Rule into the NBRS-5.0 software set, essentially making it a fixed part of the NBRS-5.0 product. The software can then be released to Regional and Headquarters Engineering Teams so that all Network Box products delivered from that time forward contain the Access Control Rule, thus protecting all FUTURE Network Box customers in a matter of SEVERAL HOURS.

Nick Jones
Head of R&D, Network Box

Nick brings depth of technology expertise to Network Box. He is an enthusiastic participant in the Free and Open Source Software community, which has led him into projects such as the Linux Kernel, OpenSSL and the ASIO networking library. He is also a contributor to the ISO C++ Standardization Committee Study Group for Networking.
The goal of SSL, or the Secure Socket Layer, is to provide authentication and encryption for a connection between two networked applications such as a web browser and a web server. Despite the fact SSL uses advanced mathematic algorithms that have been the focus of much research and analysis to implement secure encryption, the protocol can suffer from vulnerabilities in other contexts. One can think of this as a weakness in the language itself rather than a weakness in the encoding of the messages being communicated between the two endpoints.

One recent protocol vulnerability has been dubbed CRIME, an abbreviation of Compression Ratio Info-leak Made Easy, and in this article, I will provide a rough summary of the vulnerability, to give insight not only into how it works, but how a protocol vulnerability can exist, and how this kind of vulnerability is not necessarily a problem in the domain of deep mathematics but in the domain of simple logic and design.

An attacker wants to gain access to a victim's account on their favorite website. A good way for them to do this is to get access to the Browser Cookie, the security token sent back and forth between the website and the victim's browser. Although the attacker is able to directly observe the communication between the browser and the website, they are not able to directly access the Cookie because the communication is encrypted.

Inside the encrypted communication stream, the HTTP header may look something like this:

```
GET /page HTTP/1.1
Host: example.com
Cookie: secret=xyz123
Referer: example.com/home
...
```

One feature of the SSL protocol is the option of Compression. When enabled, this allows more efficient use of bandwidth between the browser and the webserver by compressing application data before encrypting it. The compression algorithms used by SSL are standard in the world of computing, whereby they inspect the data to be compressed and try to find repeating patterns. When a repetition is found, the second and subsequent occurrence of the pattern will refer to the first occurrence, thus enabling a reduction in data size, for example:

```
GET /page HTTP/1.1
Host: example.com
Cookie: secret=xyz123
Referer: example.com/home
...
```

```
GET /page HTTP/1.1
Host: example.com
> Cookie: secret=xyz123
> Referer: #!/home
...
```

This is a very simplified explanation, but sufficient for this article.

Although the attacker cannot see the content of the encrypted stream, and the valuable cookie stored therein, they are able to measure the size of the encrypted content, which they know to be compressed. The attacker has one other trick up their sleeve, and that is the ability to inject content into the data sent between the browser and the server, this can be trivially done by placing a link to an image belonging to the first website on a second website that the attacker controls and has coerced the victim to visit. Other means of doing this can be more complex, for example Cross Site Scripting, or XSS.
By one of any number of these methods, the attacker is able to subtly affect the content of the HTTP request sent by the victims browser to the secure website. This alone doesn't give the attacker access to the secure Cookie due to security mechanisms that are standard amongst all web browsers, but it does give the attacker the control they need to discover the secure Cookie.

So the attacker is now able to make the victims web request look like this:

```
GET /page HTTP/1.1
Host: example.com
Cookie: secret=xyz123
Referer: example.com/home
...
<sneaky>Cookie: secret=a</sneaky>
...
```

The format of the Cookie used by the website is not a secret, anyone can observe this if they visit the website themselves, so what the attacker does is insert the prefix of the cookie, and a guess as to the first character in the Cookie value, into the victims website request. Analysis by the compression algorithm will lead to the following breakdown:

```
GET /page HTTP/1.1
Host: example.com
Cookie: secret=xyz123
Referer: example.com/home
...
<sneaky>Cookie: secret=a</sneaky>
...
```

As expected, the attacker has triggered a compression match using their injected content and they will carefully observe the size of the encrypted data stream. The attacker expects that the 'a' character they have chosen as the first value of the Cookie is incorrect, so they will continue to induce requests between the victim and the secure website, experimenting with different guess values. When the attacker chooses the value 'x', they will observe the following compression analysis:

```
GET /page HTTP/1.1
Host: example.com
Cookie: secret=xyz123
Referer: example.com/home
...
<sneaky>Cookie: secret=x</sneaky>
...
```

Immediately the attacker will understand that they have found the correct first character of the secret Cookie, because although they made no change to the length of the request input, after all they only only changed the value of a single character, they were able to observe a change in the length of the encrypted output.

The attacker then repeats this process until all the characters of the Cookie have been identified, at which point they have sufficient information to assume the victims identity as understood by the website.

**Conclusion**

Although this is a simplified description of the CRIME attack, it provides a good illustration of how not all vulnerabilities in computer security are based on mathematic weakness, and not all secrets require a supercomputer to crack.

The commonly accepted method of mitigation for the CRIME attack is to disable the Compression extension for the SSL protocol in cases such as the above where compression is performed with such naivety. Indeed major websites such as gmail.com, github.com and dropbox.com have made this change at the server side, and browsers such as Firefox and Chrome have made this change at the client side a long time ago.

The Network Box Research and Development Team has disabled SSL Compression by default in NBRS-5.0 applications. Network Box has decided to err on the side of caution in regard to this issue, following the generally accepted consensus of the Security Community, until other methods of mitigation become available.
Network Box
Secure Socket Layer Security Strategy
(SSL Plus) part iii

SSL is the technology used by HTTPS to provide security to web browser and web application communications. As with any modern technology, SSL can suffer from security issues of its own, because of problems with its design, or flaws in its implementation. To provide the highest level of security for HTTPS, a web application servers’ SSL subsystem must be kept up to date. Unfortunately SSL subsystems are often closely bundled with their host Operating Systems, causing SSL to suffer from the same maintenance issues as Operating Systems, namely lack of long term vendor support or complicated dependencies between web application software and host Operating Systems that restrict upatability.

Network Box recognizes the complexity of maintaining SSL and Operating System software, so developed SSL Protocol Upgrade, which is one of the major features of the Anti-DDoS WAF+ system.

In addition to terminating SSL on behalf of web application servers, the Anti-DDoS WAF+ is able to upgrade a customers secure web profile by providing the latest and most up to date implementations of the SSL protocol.

The Network Box Research and Development team is dedicated to the active maintenance of the NBRs-5 SSL subsystem, providing mitigation against known vulnerabilities such as BEAST and CRIME.

As proof of this claim, Network Box installed SSL termination and Protocol Upgrade on a customer website, and with the customers agreement will use analysis of the security of the website in this article.

By importing the signed certificates for the customer domain into the nbconfig system, and applying a few simple configurations to the existing Anti-DDoS WAF+ that is already protecting the website, SSL termination and SSL Protocol Upgrade were enabled in a matter of minutes.

As a control in this demonstration, we used the original customer HTTPS web site, before enabling of the Network Box Anti-DDoS WAF+ SSL Protocol Upgrade.
Using the renowned Qualys SSL Labs analysis tool found at https://www.ssllabs.com/, we can test the difference in effectiveness between Anti DDoS WAF+ SSL Protocol Upgrade, and standard traditional HTTPS of the original unprotected website.

Firstly, the result of analysis on the original HTTPS web site:

We will now see the results of the SSL Labs analysis of the customers website when SSL termination is provided by Anti DDoS WAF+ Protocol Upgrade:

We can see that SSL Labs has discovered many issues with the standard SSL execution using termination built into the customers web application server. Although some of the issues pointed out in the results are due to the age of the SSL software on the host system, for example susceptibility to the BEAST attack and Insecure Re-Negotiation, some of the issues are actually due to the way the web application server itself configures its SSL endpoint. Although some of the configuration issues can be altered by administrators, for example the Cryptographic Cipher algorithms that the server will allow in SSL sessions, some configurations cannot, for example the use of SSL Compression, in this case leading to susceptibility to the CRIME attack.

‘A’, Clearly a satisfactory result! Using a combination of up to date SSL software on the NBRS-5 host system and the superior configurability of the SSL Protocol Upgrade feature, we were able to take the customers website from a state of having a poor SSL security implementation, to having a world class ‘A’ grade SSL profile in a matter of minutes.

Conclusion

Keeping abreast of changes in the SSL landscape and keeping a web site SSL profile up to date is a difficult task for web service administrators because SSL software subsystems are often closely tied to web application server and Operating Systems software. Network Box, using the Anti DDoS WAF+ SSL Protocol Upgrade can remove from host web application servers not only the physical runtime burden of SSL encryption, but also the administrative and maintenance complexity.

For a web application server where an up to date SSL solution is either impossible to configure, or even never existed, Network Box Anti-DDoS WAF+ SSL Protocol Upgrade can provide an ‘A’ rated public SSL profile for customers.

Note: The 90% score on Key Exchange metric is due to the 2048 bit certificate provided by the customer. If a 4096 bit certificate was used (as currently recommended), the score would be 100%.
A typical tactic for internet fraudsters is to use an authentic looking email to lead victims to a fake version of a reputable website that the victim is familiar with. The fake website is visually identical to the original, often using the original design and layout assets from the real website, so the user is fooled into submitting personal information into the fake website, namely their username and password, as there may be no visual cues in the website content that give away its inauthenticity.

One important clue, which is the fact that the SSL Certificate of this reputable website is invalid. SSL Certificate validity is an issue that reputable websites take very seriously, so the fact that such a website has an invalid certificate should raise alarm bells to the potential fraud victim, or any internet user.

Unfortunately, the urge to settle the sense of alarm instilled in the victim by the fraudulent email is strong enough to cause them to ignore the SSL Certificate Validity Warnings raised by their web browser. The victim will bypass the security warning, and carry on right into the trap of the fraudster.

With NBR5.0 Client SSL Protection, another component of SSL Security Strategy, the browser transaction at the SSL protocol layer is analyzed in addition to the standard HTTP protocol layer. This information is made available for the Network Administrator to create Security Policy rules. For example, the Network Administrator may demand that the NBR5.0 device prevent a network user from bypassing a SSL Certificate warning.

Because the NBR5.0 device is able to interpret the SSL transaction and is aware of such a violation, it will block a user attempting to bypass the certificate warning. Alternatively, the Network Administrator may allow users to carry out this action, but occurrences will be logged as non disruptive alerts that may be analyzed at a later time.

One significant outcome of the SSL Certificate bypass protections provided by NBR5.0 is that bypass decisions are moved to the Network Gateway, where the Network Administrator is able to apply security policies that are enforceable network wide, and traceable through event logging.

Monthly SSL Housekeeping

In May 2013 the Network Box platform development team released a software update for a foundational component in the NBR5.0 SSL Software Ecosystem. The purpose of the upgrade was to address a number of security vulnerabilities in cryptographic algorithms that had been discovered recently.

Because one of the goals of the Network Box SSL Strategy is to move SSL execution to the network gateway, with the Network Box SSL Protocol Upgrade feature being a classic example, an update to the NBR5.0 SSL software ecosystem is an update to the SSL ecosystem for the whole network.

In June 2013, the Network Box platform development team will again be busy updating the Root Certificate Authority Database in NBR5.0. This database contains the lists and Public Certificates of the highest level SSL authorities and it is by these authorities that the SSL Certificate of public websites must be signed, for the reason that the same authorities are also trusted by applications that use SSL, such as web browsers.

A future issue of In the Boxing Ring will go into more detail about SSL Certificate Authorities and how they are a part of the NBR5.0 SSL Security Strategy.
Java, Vulnerabilities and Exploits

A common question that Network Box SOCs around the world are asked is “I hear Java has a new zero-day vulnerability – am I protected?” For a simple question, the answer is somewhat complicated.

As a general rule, if you see some new vulnerability in the news, or there is something being actively exploited (or about to be actively exploited), then Network Box Security Response is closely monitoring it and releasing whatever protection we can. Our overriding goal is to secure our customers’ networks, and that means it is our 24x7 job to stay on top of the security threat landscape.

Java continues to have new zero-day vulnerabilities. So, for clarity, Network Box adheres to the Common Vulnerabilities and Exposures (CVE) system (cve.mitre.org) and you’ll see Network Box listed as a CVE output conformant partner on the mitre site. The CVE system allows the industry to universally refer to the same vulnerability by the same name, and is very well adopted. We only wish such a universal convention existed for virus naming.

Let’s now discuss the difference between a vulnerability and an exploit, and start with the definitions:

- **A vulnerability** is a problem that can lead to a security compromise.
- **An exploit** is something that takes advantage of a particular vulnerability to achieve that compromise.

A vulnerability could be something as simple as a buffer overflow – a system expecting say 100 bytes of input receives 1000 bytes to overflow the buffer and overwrite other parts of memory. Or, it can be a more sophisticated sequence of actions forcing the system to behave undesirably. In some cases, exploit of the vulnerability simply crashes the affected systems. But, in other cases, code can be delivered by the exploit and executed on the system. The so called ‘remote root compromise’ is the worst of these cases – and means that a system is remotely compromised to give root (aka administrative) level access to the attacker.

Protecting against particular exploits is relatively easy. Just like viruses, pattern matching and/or heuristic behavior can be used to detect the exploit and block it. But, that does nothing to protect against other exploits of the same vulnerability. An example may help:

Say a vulnerability exists in a system such that if the result of a calculation is zero the system crashes. An exploit is released, asking the system to calculate “1 divided by 0”.

We can protect against that particular exploit by looking for “1 divided by 0”. But, what about another exploit that says “1 divided by (2 minus 2)”, or perhaps “10 divided by (sin(0 radians))” or “let a=0. Then 100 divided by ‘a’”. Protecting against all possible exploits of the vulnerability is not trivial, or in some cases even possible. Often it requires complete validation and even simulation of the vulnerable environment.

In the given example, the only way to 100% protect that vulnerability is to provide a complete arithmetic system that calculates the outcome of the part of the formula after “divided by” and ensures it is non-zero.

One possible solution to the problem (used by some security products) is to be aggressive – for example detecting “divided by” in the message, and blocking that as an exploit of the vulnerability. While effective, such sledgehammer approaches often cause more problems with false positives than they solve.

Protecting against all possible exploits of a vulnerability is not impossible, in most cases, but often extremely difficult. Protecting against particular exploits is relatively simple.
So, what does Network Box Security Response do when faced with a new vulnerability? In general, we follow the following methodology:

1) We gather all the information we can get on the vulnerability – from our own research as well as from partners (such as Kaspersky, Microsoft MAPP, security groups, underground forums, etc).

2) We gather all the known exploits.

3) We first prepare and release both exploit and exploit fragment signatures. An exploit signature will match one particular exploit. An exploit fragment signature matches common parts of exploits – often with more than one fragment having to match in order to block the threat. This acts as a first-line defence and can be quickly released. In the early days of an outbreak, existing exploits are often simply used, or the hackers just re-use portions of existing exploits in their code. While not perfect, such an approach is effective in the early days of an outbreak.

4) We monitor our Outbreak real-time system to look for new exploits – in particular keeping an eye on exploit fragments and checking actual hits to ensure that we are not getting any false positives. We also feed this information back to our partners and security groups.

5) We analyse the core vulnerability itself and see if there is any general-purpose way of protecting against all exploits – either by signature or heuristic. Here, we have to balance the complexity of the defense vs risk of false positive and likelihood of success. If possible, we release such vulnerability protection.

The previous approach is very iterative, and we work closely with our partners at all stages. In the case of new zero-day vulnerabilities it is unlikely that a single signature can be quickly released to protect against all possible exploits. Instead, the approach we use is to quickly release what protection we can, and then improve that over time.

Should I Disable Java?

This too is a common question, but one that you yourselves must answer. Network Box does not dictate policy – we merely help to enforce the policy that you, the customer, set.

Java has a poor history of security, with a large number of vulnerabilities and a slow patch cycle from Oracle. While Oracle can, and has, patched new zero-day vulnerabilities relatively quickly, they often leave vulnerabilities open unpatched for too long. That latest ‘in the news’ zero-day may get patched in days, but a dozen other less publicized ones will remain unpatched for months.

So, my personal advise to you is similar to the advise from US-CERT – that you should look at the requirements for browser-based Java in your organization, and consider disabling it by default on machines that don’t require it. In the end, that decision is yours to make.
How to be a Prepper
(aka How to Survive a DDoS attack)

Distributed Denial of Service (DDoS): The unthinkable. The thing that puts terror into the heart of an IT administrator.

Preppers: Survivalists. Individuals or groups who are actively preparing for emergencies.

When asked about DDoS attacks and how Network Box can better help our customers, we always reply in the same way - “how prepared are you?” With DDoS, as with most emergency situations (such as a successful hack attack, or web site defacement), the best thing you can do is to be prepared. To think through the possibilities of such an attack, put a written plan in place as to how to respond to such an attack, and then file it away for when the unthinkable actually happens. So, here we have “how to be a prepper” (aka How to survive a DDoS attack).

Source of the Attack

Most DDoS attacks are external to your network. The internal ones are relatively easy to handle (find the culprit and shut him down), but the external ones are harder to stop (because you can't easily find the culprit, and it is very hard to shut him down when his attack is using 10,000 different machines across 100 countries). For an external attack, all you can really hope to do is (a) mitigate it (reducing the impact on the services your network provides), and (b) provide clues to identify the source to your upstream providers and (optionally) law enforcement.

External attacks can generally be divided into two classes:

1. Those that spoof the sender source addresses and try to overwhelm your incoming bandwidth or resources.
2. Those that do not attempt to spoof the sender source addresses and generally try to overwhelm your outgoing bandwidth or resources.

While the technology to defend against each type is very different, the general approach to plan for such attacks is similar.

Denial of Service by your ISP

The first step in any plan for DDoS mitigation is to talk to your Internet Service Providers (ISPs). The attack is coming in over their network on its way to attack you, and some ISPs are more concerned about their own networks than helping you. It is not unheard of for an ISP to implement upstream blocks (at their borders) for traffic destined to your network (effectively cutting you off from the Internet). If your ISP behaves like that, it does not matter what you do in your own network, your ISP is going to DoS you no matter what protections you put in place.

As an example, one popular ISP has the following stated policy:

- For a first-time DDoS, the attacked IP address will be blocked for a minimum of 1 day.
- For any subsequent DDoS, within 3 months from the date of first DDoS, the attacked IP address will be blocked for a minimum of 4 days (even if the attack has ceased).

This is the attacked IP address (ie; the victim - you), not the attacker. If you used that ISP, the first time you were the victim of a DDoS attack, you would be cut-off for 1 day. If you got attacked again within 3 months, you would be cut-off for 4 days.
So, the first step in planning for DDoS is to talk to your ISPs and find out their policies surrounding DDoS attacks. Find the ISPs that explicitly state they will work with you in resolving the situation and will not block your IP address without express permission from you.

IP Addresses – the more the merrier

The next step is to look at the IP addresses you have been assigned (or own yourself, if large enough), and what public services you offer on those addresses. Try to keep a large pool of addresses free, and keep the DNS TTL (time-to-live, expiry) records short for those services (to allow you to quickly switch IPs if necessary).

Often, DDoS botnets don’t correctly follow the Internet standards for caching of DNS records – they’ll continue to attack the same IP address long after you’ve switched to a different one.

Distribution of Services

Next, try to distribute your services. Decide those which you must keep in-house and those which can be offloaded to a different network (or hopefully multiple different networks). By spreading your services across different data centers, you can increase the likelihood that you will have some availability when under attack. Simple services such as DNS are good candidates for distribution (and being UDP based, are very susceptible to spoofed source or reflection attacks).

Resource Planning

While being happy that your web server or firewall can cope with normal traffic with only 50% utilization, that 50% free capacity is likely to disappear very quickly when under DDoS attack. You need to put in sufficient equipment to deal with attack-level requests, not day-to-day level ones.

This can be expensive, so do the calculations to determine what is a reasonable level of incoming requests, and outgoing replies, to plan for. Base that on the bandwidth you have available and the complexity of the services you expose. Then, work back from those calculations to determine what resources you need to be able to serve that amount of requests.

It is not just about the box

Planning for a DDoS attack is not just about the DDoS protection box you put in front of your network. Even the best DDoS mitigation appliances will be no good if your ISP cuts you off, or your upstream bandwidth is saturated.

The military has an adage called the 7 Ps - Proper Planning and Preparation Prevents Piss Poor Performance – adhering to such advise may just save you one day.

Once you have your plan in place, communicate it to your partners (ISPs, security and other service providers) as well as internally. Then, file it away in a place you can get to should the unthinkable happen.
The Composite Blocking List (CBL - a division of Spamhaus) has recently stepped up their actions and is now using HTTP sinkholes to try to detect trojan infected clients, so that they can then blacklist the IP addresses those clients are connecting out on. An HTTP request can get your IP address SMTP blacklisted. This is drawing attention to the problem of trojans 'calling home', as well as the monitoring, early detection, and blocking of such activity.

Network Box Security Response has recently made the following 6 recommendations that can better protect you against the threat of undetected trojans within your network:

1. Configure an effective outbound policy both in the firewall LAN->NET as well as Network Box proxies. In particular, permit only that which is explicitly required, and malicious categories such as "Virus/Malware Infected" should be blocked by default.

2. Enable the Network Box Intrusion Detection and Prevention System (NBIDPS) on your network.

3. HTTPS proxy 'CONNECT' connections to dotted decimal addresses (e.g. 123.456.789.0) should be denied by default, and only permitted where explicitly required.

4. Executable blocks (both by extension and file type) should be enabled for inbound eMail, wherever possible.

5. Different outbound proxies can be configured to use different IP addresses. Network Box generally recommends that outbound eMail be sourced from a different IP address than outbound HTTP/HTTPS traffic. Following this recommendation will mean that HTTP blocks such as CBL is now using would not affect the eMail service.

6. The service RBLMON (http://www.rblmon.com/) will monitor IP addresses you list against 60+ RBLs and alert you should one or more of your addresses by blacklisted. This service will monitor up to 3 addresses for free, and they have paid plans for larger customers. There are also several other similar competing services (rbltracker, rbl-check, rblwatch, etc). It makes good sense to subscribe to such services and keep an eye on the RBL state of your addresses. If you use a different outbound IP address for your HTTP/HTTPS vs SMTP traffic, then you should monitor both IP addresses (so that you will get notified of a HTTP/HTTPS problem even though it won't affect your SMTP traffic).

Network Box continues to closely monitor the situation, and will PUSH further protection signatures as necessary. Following the above guidelines should help mitigate any trojan infection (for example, a laptop brought into the office network) and provide for early detection and blocking of the undesirable outbound traffic.

The CBL lists IPs exhibiting characteristics which are specific to open proxies of various sorts (HTTP, socks, AnalogX, wingate, Bagle call-back proxies etc) and dedicated Spam BOTs (such as Cutwail, Rustock, Lethic etc) which have been abused to send spam, worms/viruses that do their own direct mail transmission, or some types of trojan-horse or 'stealth' spamware, dictionary mail harvesters etc.

http://cbl.abuseat.org/
Real-Time Detection and Blocking of Outbound Trojan Activity for NBRS-5.0

NBRS-5.0 has a specific framework for detecting, and policy control of, botnet infections in internal network segments as well as connected VPN networks. This is delivered as an optional security module called "Infected LAN". The framework uses three main technologies to detect anomalous internal client behavior:

1. Connections to know compromised command and control centres.
2. Signature-based and heuristic detection of network traffic from botnet clients to command and control centres.
3. Rate based and anomalous response exceptions.

If such traffic is detected from a client on the internal network, the client can be automatically isolated and administrators notified of the event.

In August, we released the base network framework for this system, and in September we followed-up with the proxy level framework.
The Open Web Application Security Project (OWASP) is an open community dedicated to enabling organizations to develop, purchase, and maintain applications that can be trusted. They have recently released their Top 10 Most Critical Web Application Security Risks for 2013. This document provides an excellent foundational understanding of the primary threats affecting web applications, and how you can best defend against them.

**LINK:** [OWASP Top 10 2013 document (PDF)](www.network-box.com)

The Network Box Anti-DDoS WAF+ system (based on NBRS-5.0 platform) is specifically designed to provide effective and comprehensive protection against these, and other, web application attacks.

*For further information, please refer to the December 2012 In The Boxing Ring article or contact your local Network Box support office.*
In August, we released our NBRS-5.0 application identification framework to public beta test.

This framework includes a proxy type called 'applicationid' capable of automatically analysing network traffic and determining the application responsible for such traffic. The system will then label the connection appropriately (for reporting and policy control). In this way, you can detect traffic such as Skype, QQ, FTP, HTTP, Facebook, and more than 1,000 other recognised applications - all based on the traffic itself and not just the network port/address it is on.

For protocols supported natively by the proxy (for example, HTTP supported by the web client proxy module), once the application has been identified, the session can be seamlessly upgraded from the applicationid proxy module to the protocol-specific proxy module. So, for example, HTTP traffic on an open port tcp/81 can be detected and moved into the webclient module for higher-level protocol control (such as authentication, policy control and anti-virus scanning).

This is a very exciting offering for us, as it provides a foundation for a lot of capabilities. Once the application has been identified, meta data can be extracted from the data stream and options for comprehensive reporting and policy control enabled.

Lite
A 'free of charge' light version, capable of identifying 10 common applications, and included with every service package with proxy services. The applications detected are our core protocols, SSL, plus several voice/video services. In addition, custom policy rules can be used to manually identify other applications using such criteria as addresses, ports, urls, etc. The 10 applications identified are:
HTTP, SMTP, POP3, IMAP, FTP (incl. FTPCTRL and FTPDATA), SSL, SIP, H.323, Facetime, GTALK (incl. GTALKAUD and GTALKVID)

Full
The full version of the system, capable of identifying more than 1,000 applications.

We will be offering licenses for two versions of this system:

www.network-box.com
In September, we announced the first releases of security modules for mail scanning in NBRS-5.0. These first modules provide for anti-malware and anti-spam scanning of SMTP eMail envelopes and messages, both inbound and outbound.

Transparency

As the NBRS-5.0 proxies are transparent (typically proxying traffic without changing the source IP address), configuration and deployment is typically very simple. Two proxy modules are provided:

- smtpclient - for protection of trusted SMTP clients, typically connecting outbound
- smtpserver - for protection of trusted SMTP servers from untrusted clients, typically connecting inbound

The primary difference between the two modules is that the smtpserver module performs third-party relay protection, while the smtpclient module does not. However, the different modules are also accounted for differently in reporting.

The smtpserver module can be configured to automatically promote a connecting client to be a trusted smtpclient, in each of two cases:

1. The connection arrives from a source IP address configured as trusted.
2. The SMTP client successfully authenticates to the trusted SMTP server.

As usual with NBRS-5.0 modules, SSL filters are configurable on both the incoming and outgoing sides of the connection, and both IPv4 and IPv6 protocols are supported (including the ability to bi-directionally translate between IPv4 and IPv6, SSL encrypted or not).

Scanning

With mail scanning in NBRS-3.0, once the entire message envelope (sender and list of recipients) had been collected, it was scanned and a single result returned for all of that sender and recipients. Similarly for the message body, a single result was returned for all recipients. If the mail was SPAM for one, it was SPAM for all.

NBRS-5.0 does this very differently. With NBRS-5.0, each stage of the message transmission is scanned individually, and individual results returned:

- When the HELO/EHLO message is received, that (along with the source IP address and other attributes) is scanned.
- When the MAIL FROM sender is received, that (along with the results and attributes from the previous HELO/EHLO scan) is scanned.
- When each RCPT TO recipient is received, that (along with the results and attributes from all previous stages of the scan) is scanned.
- When the message itself is received at the DATA stage of the scan, that (along with the results and attributes from all previous stages of the scan) is scanned.

At each stage, policy rules are run and the individual recipient, or entire message, can be blocked.

This approach gives NBRS-5.0 a very fine-grained capability to control the reception of eMail messages - minimizing bandwidth usage and maximizing the performance of the appliance. In particular, a lot of emphasis has been placed in increasing the number and capability of scanning engines at the early envelope stage (before message body reception), as well as avoiding duplication of scanning effort at both envelope and message scan stages.
Classification

It is important to point out that NBRS-5.0, at its core, is merely a classification engine. During the scanning process, it identifies aspects of the object being scanned (envelope, or message, in this case) and returns a list of classifications, confidence in those classifications, and threats identified. It is then up to the policy to be configured to determine the disposition, once the scan results have been determined.

For example, unlike in NBRS-3.0, the mail scanner in NBRS-5.0 does not quarantine mail. Instead, it merely indicates back to the proxy that a threat (malware, for example) has been discovered, and classifies the object. The policy engine in the proxy can then be used to permit or deny the message, and to quarantine as necessary. For example, a common configuration would be:

```
  config network proxy rule smtpclient deny
  isthreat = TRUE with quarantine
```

That would deny (block and log) as well as quarantine any objects found to be a threat during smtp client scanning.

Example classifications include:

- **malware** - the object is determined to be malicious
- **spam** - the object is determined to be spam
- **bulk** - the object is determined to be bulk email
- **testfile** - the object is determined to be a standardized test file

Importantly, as well as providing the classification, the scanning engine provides the confidence in its classification, expressed as a percentage. 100% would indicate the normal recommended threshold to block a threat, but configuration control can be applied to be more (>100%) or less (<100%) conservative. This approach works particularly well with the multi-engine scanning approach taken by Network Box. By allowing the confidence levels from each individual engine to affect the overall score, the engines work together to provide a clear indication of the likelihood of a particular classification.

Conclusion

The NBRS-5.0 mail scanning system enters beta testing with the September 2013 NBRS-5.0 Patch Tuesday.

File scanning in
Web Application Firewall

In September, we announced upcoming support for file scanning in Network Box NBRS-5.0 WAF+. This optional facility allows files being uploaded to your websites to be scanned for potentially malicious content by the NBRS-5.0 anti-malware systems. In addition, as a unique feature, you can also optionally enable scanning of outbound traffic (perhaps not for the entire website, but only for certain paths). Support for this functionality is currently under trial on several sites, and will be formally released in the October patch tuesday.
IDS/IPS
versus
WAF

Since the launch of Network Box NBRS-5.0 Anti-DDoS
WAF+ we’ve received many enquiries asking about
the differences between traditional IDS/IPS systems
and the Network Box Anti-DDoS WAF+ offering. Both
protect web servers, so which is better?

This article answers this question

IDS/IPS
Under the Hood

Intrusion Detection / Prevention Systems operate at the network level, usually in a transparent manner. They receive network packets,
and attempt to recreate the communication sessions that those packets make up. They can then provide advanced higher-layer
decoding, for specific protocols, but they always operate at the network layer itself.

At the packet level, let’s see how this works with a typical HTTP session over tcp/80 destined to a web server. Separating out just the
one session we are interested in, this is what the IDS/IPS system seems at the network packet layer:

09:35:57.858874 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [S], seq 893507093, win 65535, options [mss
1460, nop, wscale 4, nop, opt, T5 val 3712851838 ecr 0, sackOK, eol], length 0
09:35:57.862643 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [S.], seq 2557492822, ack 893507094, win 42900, options
[mss 1430, nop, sackOK, nop, wscale 6], length 0
09:35:57.862817 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], ack 1, win 16384, length 0
09:35:57.862873 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [P.], seq 1,46, ack 1, win 16384, length 145
09:35:57.866990 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], ack 146, win 669, length 0
09:35:57.881756 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 1,1431, ack 146, win 669, length 1430
09:35:57.881881 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 1431,2861, ack 146, win 669, length 1430
09:35:57.881883 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 2861,4291, ack 146, win 669, length 1430
09:35:57.881904 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 2861, win 16205, length 0
09:35:57.881916 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 2861, win 16384, length 0
09:35:57.882176 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 4291,7521, ack 146, win 669, length 1430
09:35:57.882520 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 7521,7151, ack 146, win 669, length 1430
09:35:57.882522 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 7151,8581, ack 146, win 669, length 1430
09:35:57.882271 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 146, win 16205, length 0
09:35:57.882325 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 8581, win 16384, length 0
09:35:57.882499 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 8581,10011, ack 146, win 669, length 1430
09:35:57.882532 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 10011,11259, ack 146, win 669, length 1248
09:35:57.882550 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 11259, win 16216, length 0
09:35:57.883065 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 146, win 16384, length 0
09:35:57.886031 IP 74.125.128.147.80 > 10.8.2.100.65132: Flags [.], seq 11259, win 16384, length 0
09:35:57.886067 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [.], seq 11260, win 16384, length 0
First, it sees a SYN packet going 10.8.2.100 -> 74.125.128.147, followed by a SYN+ACK coming back, followed by an ACK - this is the classic three way handshake and shows a successful TCP/IP connection. Then, it sees TCP/IP data travelling back and forth, along with packet acknowledgements, and possibly re-transmissions, duplicate packets, and all the other mess that occurs at the IP layer. Finally, it sees a FIN packet going 10.8.2.100 -> 74.125.128.147, followed by a FIN+ACK and finally an ACK - this is the TCP/IP connection being closed.

A good IDS/IPS will start at the network layer, and try to re-assemble the packets into streams (TCP/IP streams in this example). Once it has the stream, and has been told the protocol involved, it will attempt to decode the protocol data. Let's have a look at the fourth packet, in detail:

```
09:35:57.862873 IP 10.8.2.100.65132 > 74.125.128.147.80: Flags [P.], seq 1:146, ack 1, win 16384, length 145
0x0000: 4500 00b9 f888 4000 4006 0000 0a08 0264 E.....@.@......d
0x0010: 4a7d 8093 fe6c 0050 3541 d616 9870 3e57 J}...l.P5A...p>W
0x0020: 5018 4000 d827 0000 4745 5420 2f20 4854 P.@..'..GET./.HT
0x0030: 5450 2f31 2e31 0d0a 5573 6572 2d41 6765 TP/1.1..User-Age
0x0040: 6e74 3a20 6375 726c 2f37 2e32 342e 3020 nt:.curl/7.24.0.
0x0050: 2878 3836 5f36 342d 6170 706c 652d 6461 (x86_64-apple-
0x0060: 7277 696e 3132 2e30 2920 6c69 6263 7572 rwin12.0).libcur
0x0070: 6c2f 372e 3234 2e30 204f 7065 6e53 534l/7.24.0.OpenSSL
0x0080: 2f30 2e39 2e38 7820 7a6c 6962 2f31 2e32 /0.9.8x.zlib/1.2
0x0090: 2e33 3e0a 486f 7374 3a20 743a 2e33 3235 .3:.Host:.74.125
0x00a0: 3e31 3238 3e31 3437 0d0a 4163 6372 7572 >1.28>1.47..Accept
0x00b0: 7277 696e 3132 3e0a 0d0d 0d0d 0d0d rwin12.>....
```

At the start of the packet is the ethernet, IP and TCP encapsulation. The data itself starts at offset 0x0028 and contains the following:

```
GET / HTTP/1.1
User-Agent: curl/7.24.0 (x86_64-apple-darwin12.0) libcurl/7.24.0.0.OpenSSL/0.9.8x.zlib/1.2.5
Host: 74.125.128.147
Accept: */*
```

That is an HTTP request, and the request, headers and values can be decoded by high level protocol decoders in the IDS/IPS.

The problem for most IDS/IPS systems start here at the application layer, and that is why most IDS/IPS systems stop here. The encodings, and data transfers, at the application layer become very complicated, so traditional IDS/IPS systems generally decode only up to, but not into, the application layer. An IDS/IPS system can see the application traffic, but can't understand it. IDS/IPS systems just blindly apply pattern matching signature rules against both raw packet and stream re-assembled data, but they do not understand the meaning of the data itself.

### WAF

#### Under the Hood

A WAF, such as Network Box NBR5-50 Anti-DDoS WAF+, starts where the IDS/IPS finishes. Web Application Firewalls decode the HTTP web protocol completely, and understand the meaning of requests and responses on that protocol.

Examples of this include:

- A WAF typically takes the stream level data, and correlates streams together (so for example, requests to the same web application from the same web client, are treated together).
- WAF systems can protect both HTTP and HTTPS protocol traffic.
- Web-specific formats such as HTML, JSON and XML have specific protection modules in the WAF.
- Unlike protocols such as SMTP, POP3, FTP, the HTTP protocol is used to deliver applications - and the WAF understands both that delivery process and the applications themselves.
- Web requests can include both headers as well as bodies, and the WAF understands and decodes these completely. As well as decoding headers, the WAF can decode fields and attached files within POSTed forms.
- Requests and bodies can be encoded in a myriad of formats, and the WAF is able to fully decode all these.
- Bodies can be subjected to more in-depth analysis (such as anti-virus scan).

For example, let's examine a web request that recently came in to www.network-box.com from a host in the Netherlands. Here is the request:

```
POST /phppath/php?-d+allow_url_include%3Don+-d+safemode%3Don+-d+safe_mode%3Don+-d+suhosin.simulation%3Don+-d+disable_functions%3D %22%22+-d+open_basedir%3Dnone+-d+auto_prepend_file%3Dphp%2F%2Finput+-n HTTP/1.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; tr-TR) AppleWebKit/533.20.25 (KHTML, like Gecko) Version/5.0.4 Safari/533.20.27
Content-Type: application/x-www-form-urlencoded
Host: www.network-box.com
Content-Length: 2634

(2,634 byte HTML form provided, but not shown)
```

www.network-box.com
The WAF decoded not just the headers, but also the attached form and even the fields within that form. It found the following laundry list of problems:

1. **PHP Injection Attack** - OWASP/AppSensor/CIE4
   OWASP_TOP_10/A1 OWASP_TOP_10/A6 PCI/6.5.2
   WASCTC/WASC-15 WASCTC/WASC-25
   WEB_ATTACK/HTTP_RESPONSE_SPLITTING
   WEB_ATTACK/PHP_INJECTION

2. **SQL Comment Sequence Detected** - OWASP/AppSensor/CIE1
   OWASP_TOP_10/A1 PCI/6.5.2 WASCTC/WASC-19
   WEB_ATTACK/SQL_INJECTION

3. **SQL Hex Encoding Identified** - OWASP/AppSensor/CIE1
   OWASP_TOP_10/A1 PCI/6.5.2 WASCTC/WASC-19
   WEB_ATTACK/SQL_INJECTION

4. **SQL Injection Attack: SQL Operator Detected** - OWASP/AppSensor/CIE1
   OWASP_TOP_10/A1 PCI/6.5.2 WASCTC/WASC-19
   WEB_ATTACK/SQL_INJECTION

5. **Blind SQL Injection Attack** - OWASP/AppSensor/CIE1
   OWASP_TOP_10/A1 PCI/6.5.2 WASCTC/WASC-19
   WEB_ATTACK/SQL_INJECTION

6. **SQL Character Anomaly Detection Alert - Repetative Non-Word Characters**

7. **Restricted SQL Character Anomaly Detection Alert - Total # of special characters exceeded**

8. **Detects MySQL comment-/space-obfuscated injections and backtick termination** - WEB_ATTACK/SQI

9. **Detects SQL benchmark and sleep injection attempts including conditional queries** - WEB_ATTACK/SQI

10. **Detects basic SQL authentication bypass attempts 2/3** - WEB_ATTACK/SQI

11. **Detects MySQL comments, conditions and ch(a)r injections** - WEB_ATTACK/SQI

12. **Detects classic SQL injection probings 2/2** - WEB_ATTACK/SQI

13. **XSS Attack Detected**

14. **IE XSS Filters - Attack Detected**

A total of 14 suspicious attributes in the web request headers and body.

While an IDS/IPS system blindly blocks/permits based on a single result, the WAF can operate an Anomaly Scoring System - similar to how anti-spam systems work, the WAF assigns each test a score - and the entire transaction is permitted / denied based on the total anomaly score. These tests can be done at each of the four stages of the web transaction (request header, request body, response header and response body).

In the case of the above web request, the anomaly score totalled 158 (54 being SQL injection types, and 25 being XSS cross-site-scripting types), and the request was blocked as it exceeded the default threshold of 10 (by more than 15x).

**Conclusion**

So, what is the difference between IDS/IPS and WAF? Well, simply put the IDS/IPS system operates between the network and application layers, while the WAF operates purely at the application layer. While IDS/IPS systems can provide limited support for application layer protocols, WAF concentrates purely on protection of web protocols and the applications running on those protocols.

An IDS/IPS can provide protection for a large number of protocols, while a WAF provides complete and comprehensive protection for applications running on just one protocol - the HTTP web protocol.
Network Box 5 (NBRS-5.0) is Next Generation Managed Security. It takes our core foundational UTM+ (Unified Threat Management) appliance and services, along with ten years of Intellectual Property and experience, into a new enhanced platform, which expands beyond traditional UTM+ capabilities.

In November, Network Box officially launched the new managed security platform Network Box 5 (NBRS-5.0). The new platform has been optimized and performance has been enhanced to be eight times faster than the previous (NBRS-3.0) platform. In addition, numerous additional and improved technologies have been added to the platform, including:

- Entity Management
- Application Identification
- Holistic
- SSL Proxy
- Anti-DDoS
- Web Application Firewall
- HTML-5 Dashboard
- Unified Configuration
- Transparency
- 802.1D Bridging
- Clustering
- IPv4/IPv6 Bridging
- Protocol Translation
- Multi-Core
- Distributed Scanning
- SSL Policy Enforcement
- Parallelized Scanning
- Unified Logging
- Configuration Audits
- Revision Control
- Bi-Directional Synchronization
- Device Provisioning
- Data Export
- Layer 7 Policies
- Web Sessions
- and many more...

During the official launch event, in keeping with the Network Box 5 theme, five key technologies were highlighted:

**Anti-DDoS**

**WAF+**

Unlike many dedicated Web Application Firewall systems on the market, the Anti-DDoS WAF+ includes a wide range of capabilities to allow for the mitigation of Distributed Denial of Service (DDoS) attacks.

It utilizes the OWASP top ten as a minimum set of guidelines which need to be adhered to.

The defensive strength of the default settings are maximized, and augmented by the power and accuracy of a state-of-the art real-time automated threat fingerprinting engine. This ensures as many incoming threats can be blocked as possible, using the minimum of effort; brute force protection is built-in as standard.

In the event of a newly discovered web application vulnerability being revealed, or an actual attack taking place, the Anti-DDoS WAF+ allows for the real-time installation of emergency virtual patches at the gateway, in the form of specific WAF rules, to immediately detect and prevent any specific security issues.

- Real-Time Automated fingerprinting
- Slows down attacks by a factor of up to 1,000
- Millisecond response to brute force attacks
- High performance rules engine capable of millions of rule-checks per second
- Up to 15,000 fully analyzed transactions per second
- Support for major Content Management Systems including: Joomla, Drupal, and Wordpress
Entity Management

Network Box 5 offers a revolutionary Entity Management system, which completely redefines how users and machines are monitored and protected.

Entities are Users and Devices that are parts of your network. Network Box 5 tracks these entities, their attributes (such as MAC addresses, IP addresses, email addresses, etc.), and the network resources which they utilize.

Rather than presenting a set of disparate screens, showing firewall blocks by IP address, URL access by authenticated user, and email by email address, the Network Box 5 platform presents a single holistic view of the activity, of each of the entities in your network. For example; calling up user "Joe," will show all his firewall blocks, web accesses, network usage, email, etc.; across his desktop, laptop, phone, tablet, and remote VPN.

The entity model itself is built and maintained by automated systems, and is an extremely efficient, and effective, technology, for helping IT Managers to monitor, manage, and protect, their users and networks.

- Presents a single holistic view of the activity of each of the entities in your network.
- Built and maintained by automated systems.
- Allows IT Managers to monitor, manage, and protect, their users and networks.

Secure Socket Layer (SSL) Proxy

SSL is the technology used to provide security to web browser and web application communications. However, as with any modern technology, SSL can suffer from security issues of its own, because of problems with its design, or flaws in its implementation.

Regardless, the very fact that SSL data streams are encrypted, means that without SSL-Proxy technology, it is all but impossible to scan such data streams for viruses, worms, spyware, and other undesirable content.

- Provides security to web browser and web application communications
- Allows certificate validation policy to be performed and enforced at the gateway
- SSL traffic is identified, decrypted and then subjected to security functions

Application Identification

While traditional firewalls block protocols, and ports, Application Identification looks at the traffic up to and including layer 7 to identify the application responsible for that traffic.

Once identified, policy control can be applied to that traffic. The traffic can also be ‘promoted’ – for example HTTP traffic detected on ports other than tcp/80 can be promoted to be handled by the web client scanning modules for anti-malware and policy control. Integrated to the SSL proxy, even traffic inside encrypted SSL sessions can be identified and controlled.

The Network Box 5 application identification engine, comes in both Lite and Full versions. The Lite version is included as standard with every UTM+ system; while the optional Full version is available as an upgrade. The new Application Identification system allows connections to be appropriately labelled for reporting and policy control.

- Encrypted SSL sessions can be identified and controlled
- Layer 7 traffic analysis
- Policy control can be applied to traffic

HTML-5 Dashboard

The Graphical User Interface has been updated with a state-of-the-art customizable HTML-5 Dashboard.

The new Dashboard offers IT Managers highly intuitive visual feedback, that is very easily understood and acted upon.

Due to HTML-5 being compatible with almost every mobile device on the market, the information conveyed by the dashboard is not only real-time in nature, but can be seen from almost any mobile device.

The system is also capable of automatically generating highly customizable Adobe PDF format reports. All users need to do, is create a dashboard screen showing the information they require, then tell the system to generate a report, or regular reports, matching the timeframe required.

- Real-Time Attack Monitoring
- Supports almost any Mobile Device
- Customizable Dashboards

www.network-box.com
For the release of the Network Box 5, the hardware product lineup has undergone a complete makeover. From the S-Series, to the E-Series, all models have received extensive re-evaluation, transitioning to all new hardware platforms that not only provide an increase in performance, but also make a revolutionary jump in terms of key technologies such as CPU and RAM architecture.

Overview

The new Network Box 5 hardware lineup brings much of the technologies that were previously present only in the Enterprise grade products, down to all product levels. Technologies such as 64bit, Multi Core CPU architectures, DDR3 RAM and multi queue network devices. These technologies are now standard in S-Series and M-Series for the first time, and not just reserved for the E-Series only. What this represents is a unification of the entire Network Box product line that is unprecedented in the company’s history, where the difference between each class of hardware is minimized from the point of view of the Software.

Additionally, by synchronizing the re-evaluation of all models, Network Box 5 is able to go to market on hardware platforms that all have a similar lifetime of manufacture, reducing the mixture of old and new that existed during the course of previous Network Box product cycles.

In terms of raw features, all Network Box 5 models make a significant step over previous generation models. Overall highlights include:

- Increased CPU speeds, with all models supporting 64bit multi core CPUs
- Minimum of 4GB of RAM in ALL models
- 4,000,000 hour MTBF solid state drives as primary system storage
- Minimum of 1TB secondary storage capacity for all rack mounted models
- Network controller chipsets featuring multiple hardware transmit and receive queues

www.network-box.com
S-Series

- S-35i
- S-95i

For the first time, 64bit comes to the S-Series, completing a transition to 64bit computing that Network Box was one of the first vendors in the world to start almost nine years ago.

The S-Series also comes with 4GB of RAM, a quantity previously only reserved for the E-Series. This combined with the 64bit, dual core Atom CPU, moves the performance of the S-95i into an exciting new level.

M-Series

- M-255i
- M-295i
- M-385i
- M-395i

With the M-Series, Network Box aims to provide for a wide range of performance and cost considerations, from the M-255i, which is an extension of the cost effective S-Series hardware but in a rackmounted configuration with expanded storage capacity, to the M-295i and M-385i which provide different outlooks on the price/performance sweet spot where Network Box has seen the most positive reception from the market.

In addition, Network Box is offering the new M-395i, which ventures into near Enterprise CPU performance and RAM and disc capacity, but without redundancy, in a 1U rack mountable configuration.

E-Series

- E-1000i
- E-2000i
- E-4000i
- E-8000i

The E-Series hardware lineup transcends the definition of mere Network Appliance, into the realm of High Performance Computing. Each of the E-1000i, E-2000i and E-4000i offers dual socket, multi core CPU configurations, and supports high performance 8 Transmit/Receive queue network adapters.

The new E-Series supports massive RAM capacity: 8GB, 16GB and 32GB for each of the models respectively, and disk capacities of 4TB by default, optionally expandable to 16TB by means of a hardware RAID addon card.

For specially negotiated orders, Network Box is offering a new E-Series model: The E-8000i, which comes with a massive 16 core CPU configuration, and RAM capacity up to 128GB.

Conclusion

The Network Box 5 hardware product lineup is the culmination of years of planning and development. It is as important a feature of the Network Box product line as any of the well documented software features such as Anti-DDoS WAF+ or IPv4-IPv6 bridging. It has a very close bi-directional relationship with the Network Box software feature set, influencing a number of design decisions during the development of the software; and conversely, decisions made during evaluation of the hardware platform were driven by features planned for the software.

Exciting times are surely ahead for Network Box, its worldwide distributors and its customers.

www.network-box.com
Network Box 5 includes a SSL Proxy security module that provides for identification, decryption, encryption, certificate validation, and protection of SSL network traffic. This article describes, in detail, how this proxy operates.

Certificate Exchange

The first stage of SSL consists of certificate exchange using asymmetric cryptography (usually RSA). Typically, a client connects to an SSL enabled server, and sends a message to the server to initiate the SSL communication link. The server responds with it’s certificate. This certificate is, at its core, a cryptographically signed object containing the name of the server, the dates the certificate is valid, and other optional information. The certificate is signed by an authority that both the client and server trust. For example, say a client wants to securely connect to WWW.ACME.COM - it first looks up the IP address, makes a tcp/ip connection, then initializes SSL. The server provides its certificate (which contains the name WWW.ACME.COM and is signed by a certificate authority trusted by both the client and the server) and the client validates that certificate. It is this process that provides the core foundational security of the SSL protocol. Once the secure link has been negotiated, the rest is just bulk symmetric encryption and usually very fast and relatively simple.

The vast majority of security problems with SSL come at this certificate exchange stage. Who determines which certificate authorities are trusted? Who says what to do if a server certificate does not validate (for example, it is signed by an untrusted authority, or has expired)? The answer to these questions is usually the end-user, and study after study has shown that end-users will click whatever they need to click to get access to what they want. If an end-user goes to a website, and gets a certificate validation warning, in the vast majority of cases they will just click whatever they need, to get access to the site they want.

The other issue with SSL communications is that they are encrypted, so cannot normally be subject to policy control, anti-malware or other such protection mechanisms.

So, SSL (aka TLS) is a protocol used to protect network traffic. It is commonly used as an option in such protocols as HTTP (HTTPS), SMTP (SMTP), IMAP (IMAPS), etc. The protocol is either used directly over a dedicated port (e.g.; tcp/443 for HTTPS, tcp/993 for IMAPS) or enabled mid-protocol (e.g.; CONNECT for proxied HTTP, STARTTLS for SMTP).

By convention, we refer to SSL/TLS by its original name SSL in the rest of this article (noting that both protocols are covered by the Network Box SSL proxy).

Wikipedia
Man in the Middle

The issue comes if an attacker can position themselves between the client and the server. When the client connects to the server, the attacker then intercepts that connection and provides his own fake certificate to the client. The attacker can then connect to the server as a client themselves. If the client chooses to accept the attacker’s certificate (ignoring the trust warnings raised), from then onwards the attacker can decrypt, inspect and modify all the traffic between the client and the server.

Network Box SSL Proxy

The Network Box SSL proxy attempts to solve these issues in three ways:

1. The validation and enforcement of policy regarding SSL certificates is moved from the client to the proxy.
2. Weak SSL options are removed from the SSL negotiation, ensuring that the proxy to server connection is not susceptible to attacks such as BEAST, CRIME, BREACH, RC4, etc.
3. Providing the option to decode SSL traffic for inspection, policy control, anti-malware and other such protection.

The SSL proxy works by intercepting the traffic between clients and servers, in a similar way to a man-in-the-middle attack. Application Identification is used to examine the client traffic, identify the initial SSL negotiation, and mark the connection as SSL protocol. At that point, a policy decision can be made whether to (a) bypass, (b) decode or (c) deny the connection, and this policy decision can be made with a large number of flexible access control rules (such as SSL host being visited, content filtering category, IP addresses, as well as attributes such as HTTP host for the web client protocol).

For protocols such as SMTP and HTTP proxies, where the protocol can be promoted to SSL (STARTTLS, CONNECT, etc), this policy decision can be used not just to decode the SSL traffic, but also to enforce the use of standards-conformant SSL protocol in subsequent communications (stopping clients from using CONNECT to indicate an SSL connection, but then passing non-SSL traffic over that connection attempting to bypass policy control).

The SSL proxy contains a SSL Certificate Authority, and to avoid warnings on client workstations, that certificate authority should be installed as trusted on client devices. From that point on, trust enforcement of certificates is moved from the client to the Network Box proxy. Trusted certificates will be re-signed and appear to client browsers as validated and ensured by Network Box directly.

Conclusions

Using the Network Box SSL proxy, administrators can impose effective policy control over the negotiation stage of SSL (which negotiation options are supported, which certificate authorities are trusted, which sites should be bypassed, etc), as well as choose to enforce/ bypass SSL decoding using a rich set of access control rules. Decoded traffic is subject to the same policy control and protection (such as anti-malware) as plaintext traffic.
Focus on the
NEXT ATTACK
(Not just the last one)

by Michael Gazeley
Managing Director, Network Box

At the start of each new year, the same question inevitably gets asked of me by both the media, and potential Network Box clients alike, "what should we focus on this coming year, in terms of cyber security?" Usually, the questioner then goes on to frame the question with a relatively recent marketing term or two, such as Advanced Persistent Threat, Zero Day Vulnerability, or Ransomware.

Well, ice hockey legend Wayne Gretzky famously said, "I skate to where the puck is going to be, not where it has been." This philosophy helped Gretzky become one of the most successful sportsmen of all time. And it's hard to find a better philosophy to apply to cyber security as well, for fairly obvious reasons.

Yet the vast majority of information technology managers around the world, and even well known cyber security professionals, seem to be almost pathologically fixated on the 'last attack,' rather than the 'next attack.' The last attack, either being what their own network suffered recently, or what is currently the hot cyber attack topic in the mainstream press.

For example, if a serious DDoS (Distributed Denial of Service) attack occurs against a local government department's website, making newspaper headlines, one can essentially guarantee that in the following weeks and months, IT Departments up and down the land, will be putting out Requests For Proposals for Anti-DDoS systems, arranging vendors to come in to give seminars on Anti-DDoS technologies, and generally shifting the entire focus of their cyber security onto this one single issue - until of course another kind of attack occurs - then suddenly the new attack will become the centre of attention.

Network Box's philosophy has never been to just focus on the last attack, nor do we just focus on one vector of attack. This is because your computers, networks, devices, data, and users; are probably not going to be attacked by what exactly attacked them last time; nor will they be attacked by just a single threat vector. Cyber attacks are constantly evolving, and successful cyber attacks seldom rely on just one technology or methodology, anymore.

Our Anti-DDoS technology for example, has won multiple awards right across the globe; but it is not the only aspect of cyber security which we focus on. Indeed, at Network Box, we pride ourselves on developing the very best cyber security technologies, to handle everything from zero day viruses, mobile malware, web application vulnerabilities, phishing spam, hackers, and a variety of other threats.

Just as a bodyguard in the physical world needs to be able to protect clients from all forms of harm; be it being shot, knitted, poisoned, strangled, drowned, burnt, blown up, or hit by a car; cyber security systems need to be able to deal with a very wide spectrum of threats too. And the ability to deal with all of these potential threats, also needs to be augmented with both real-time PUSH update technology, as well as twenty-four hour security operations centre monitoring, management, and support, to internationally recognized and certified standards.

So in the coming year, don't just focus on the last attack that impacted your network, nor just the current attacks which are in the headlines right now. And don't just focus on one or two threat vectors. The best way to defend your systems is to ensure every threat vector is taken into account. Because just about the only thing one can be certain of, in the realm of cyber security, as with most areas of life, is that there are very few certainties.
Firewall
Intrusion Detection and Prevention (IDP)
Virtual Private Networking (VPN)
Anti-Malware
Anti-Spam
Anti-Spyware
Web Proxy
Content Filtering
Data Loss Prevention (DLP)
Company Policy Enforcement / Compliance
Real-Time updates with PUSH Technology
Secure 24 x 7 x 365 Monitoring
ISO 9001 / 20000 / 27001 Certified Management
IPv6 Ready Core Phase-2 Certified
In-the-Cloud Protection
Comprehensive Adobe PDF Format Reporting
Apple iPhone / iPad HD Management Platform
Anti-Distributed Denial of Service
Web Application Firewall
IPv4-IPv6 / IPv6-IPv4 Bridging
Multiple Internet Connections
High Availability / Load Balancing
Internet Acceleration
Secure VoIP (Voice over Internet Protocol) Gatekeeper
Secure Video Conferencing Gatekeeper
Quality of Service Control Traffic Policing
Denial of Service Protection
Threshold Limiting
Hardware Fault Tolerance
Live Watch Real-Time Monitoring
Adobe PDF Report Generation
SSL (Secure Socket Layer) Virtual Private Networking
Anti-SPAM Pre-Scanning, bandwidth protection
Enhanced Image SPAM protection, including Optical Character Recognition technology
Mail Portal System, End User email management including SPAM release and white / black listing
Enhanced GUI (Graphical User Interface)
Secure Socket Layer (SSL) Proxy Application Identification
Entity Management
HTML-5 Dashboard