Global Information Assurance Certification Paper

SMTP Gateway Virus Filtering Using Trend Micro's InterScan Messaging Security Suite

GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4b, Option 2 – Case Study in Information Security

Finn Westerman

Abstract

This paper details a case study of an implementation of Trend Micro's InterScan Messaging Security Suite (IMSS – version 5.1), to effectively filter viruses, spam, unwanted attachments and message content in an SMTP gateway environment.

Large message volumes were filtered effectively, with the detection and removal of thousands of viruses, unwanted attachments, and spam messages on a daily basis. Although not total in the case of spam, filtering was still achieved to a high degree. Alerting and logging were also found to be useful for troubleshooting and identifying external virus outbreaks or service problems.

In conclusion, it was found that although the IMSS product was not a panacea for SMTP gateway filtering, it proved to be an extremely effective and flexible virus and spam filter, both proactive and reactive, as well as a more than useful SMTP security policy enforcement tool. It reduced many risks that were present in a previously unprotected gateway, whilst introducing relatively few new ones, or other complications.

Contents

Before

Effective anti-virus filtering at the Internet messaging gateway is a crucial component of a layered security strategy to most modern organizations, particularly as their exposure to the Internet and reliance upon connectivity to it increases. With the enormous growth in e-mail usage through the 1990's and onwards, it has become more and more business-critical. Just as an effective content-filtering solution for Internet browsing traffic can prevent malicious external threats, an SMTP gateway filter can provide an additional layer between users and the Internet, generally preventing the receipt and transmission of viruses, spam, and unwanted message content through e-mail.

Many of the successful modern examples of viruses and worms have propagated so effectively primarily due to the use of e-mail. Examples are readily available in the current virus alert lists of several anti-virus software vendors, such as Sophos' "Top ten viruses" page (http://www.sophos.com/virusinfo/topten/), and Trend Micro's "Top Threats" (http://www.trendmicro.com/vinfo/default.asp?sect=TT), where variants of 'BugBear', 'Sobig', 'Klez', and 'Lovgate' – all predominantly mass-mailing in their propagation methods – make up the majority of the list. Although typically not as fast spreading as worms that automatically exploit vulnerabilities in code and self-propagate, mass-mailing viruses have the ability to infest systems worldwide in minutes, thanks largely to the actions of unsuspecting users and their propensity to open known or unknown content from known or unknown sources. It is often the file system exploitation component of a virus or worm such as 'Lovgate' that allows it to spread rapidly once inside an organization. For example, according to Trend Micro, WORM_LOVGATE.F does this by "dropping copies of itself to shared folders with read/write access" (http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_LOVGATE.F). However, it is the mass-mailing capability that more often than not allows the virus or worm to reach the internal network in the first instance. Whether it is due to poorly configured gateways, unwitting users, or un-patched systems, the fact that major outbreaks can arise in a very short period of time means that perimeter network protection is essential in preventing and dealing with them.

As covered by the SANS Security Essentials course, a Defense In Depth strategy is important in maximizing protection of an organization's information. An SMTP gateway filter is usually the first point at which e-mail can be examined as it enters an organization. It aids in preventing internal network congestion, and to a degree, denials of service that may occur from the increased traffic resulting from a worm or virus outbreak. This is possible because the gateway resides at the network perimeter, and can hopefully bear the brunt of most e-mail borne attacks originating externally, thus saving internal layers from exposure.

The problems that arise when SMTP gateway traffic is not filtered for malicious or unwanted content, are due to the fact that it allows this content to reach the internal messaging system, or worse, the users e-mail client,

without inspection. The corresponding risk of the content being viewed or executed, and a damaging virus outbreak occurring, increases dramatically with every layer it bypasses, hence the need to filter as soon as possible, i.e. at the gateway. The increasing costs that virus incidents are found to cause organizations are shown in the ICSA Labs 8th Annual Virus Prevalence Survey, where it is stated “The average reported cost for a disaster this year was $81,000 (median $9,500) US versus $69,500 (median $5,500) US in 2001.” (Bridwell, p.45). It is therefore becoming more important to prevent viruses getting a foot in the door via an inadequately protected SMTP gateway because the resulting outbreak cleanup costs may be debilitating. The real cost is often very difficult to measure, because delays are also caused for ongoing business projects due to system and staff unavailability. Data may be lost permanently if adequate backup strategies are not in place, and therefore total recovery may not be possible.

This paper details an implementation of Trend Micro’s Internet Messaging Security Suite (IMSS) for an SMTP gateway system, and its use to effectively enforce a Defense In Depth anti-virus strategy, adding an important layer of anti-virus protection, along with other benefits for the organization. Spam issues are also dealt with, with mixed success, as is the filtering of offensive content and over-sized or unwanted attachments.

During

It was identified that the existing SMTP gateway solution presented a problem in relation to viruses and other unwanted message content, because the internal messaging systems were inundated and struggling to cope on occasions with mass-mailing virus outbreaks (for example, the 'LoveLetter' VBS worm). Although housed in a DMZ environment, with firewalls between the SMTP server and both the Internet and the internal production network, the solution simply forwarded messages between the Internet and internal messaging systems without adding any real defense at that layer against malicious or problematic content.

The existing SMTP gateway architecture comprised a primary server and a contingency server. Connectivity to these servers from the internal messaging systems could be swapped via a manual configuration change. Incoming e-mail (from the Internet) would be delivered to the primary server if available, due to it being the primary MX record in DNS, or if unavailable, the contingency server (secondary MX). Outbound e-mail could only be sent to one of the servers, because the internal messaging system was only able to connect to a single host at one time.

My role in the problem identification and solution was as project lead and designer, with assistance for testing and configuration discussion provided by other members of my support team (2 additional staff). Other support teams provided additional assistance for physical server relocation and communications setup.

To apply IMSS, the contingency server was installed first, allowing existing services to continue during the implementation with minimal disruption. During installation, external connectivity to the server was disabled to prevent any e-mail loss or excessive queuing. Following verification of the backup of the previous server implementation, the server was set up again from scratch, with Windows 2000 applied, including Internet Information Services 5 (IIS, required for remote administration and configuration), then Service Pack 3 and the latest available security hotfixes from Microsoft. Even though connectivity limitations imposed via the external firewalls meant that exploitation of IIS and other Windows 2000 vulnerabilities would be highly unlikely, latest patch levels were applied to maintain a strong baseline for both security and potential support-related issues. The SANS Security Essentials course covers security baselining, and its importance in a security strategy such as the approach used in this solution. IMSS was then installed from notes created during testing of the application in a development environment with comparative connectivity. Version 5.1 of the application was applied initially.

Despite the ability to perform installation remotely, given sufficiently privileged account credentials, it was found that setup was best performed directly on the server in the DMZ environment, due to the limited connectivity with the production network. This also reduced the chance of the installation being

corrupted due to any network interruptions, and as such would be a practical approach in general, regardless of environment.

The full version of the eManager for SMTP filter was also selected during installation, and because the server was Internet-facing, DNS was selected as the means to resolve destination hosts.

The IP address of the internal messaging system was entered for use by IMSS to send notification messages, via the standard SMTP port (25). An e-mail address for a shared internal mailbox was used and would be recommended to receive the alerts, as it was found to be the most efficient method for dealing with them because several support staff were able to monitor it simultaneously.

Setup then required administrator credentials to be entered, as well as paths for installation. A separate location to the OS drive was used for the installation path, to allow use of a RAID 5 disk setup for faster e-mail processing, recommended in Trend Micro's Knowledge Base: "RAID 5 (i.e. striped with parity) is the best configuration for email servers" (http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=13668).

Following installation, configuration was performed via a browser interface. Changing the default password for access to the GUI was an important first step, to prevent anyone other than the legitimate support staff from viewing and changing settings. There were two main areas of configuration to complete, labeled 'Configuration' and 'Policy Manager'. Key settings for the 'Configuration' were as follows:

Configuration→Security→Security Settings

- Limit compressed layers scanned to 10, and decompressed file size to 131071Kb. It is highly unlikely that any legitimate attachments would exceed these limits, which the server could handle in testing. However, not enforcing them could allow maliciously coded compressed files such as '42.zip', a "ZIP archive, 42K, composed of nested zips (nested 6 levels deep, each level 17 wide) - produces a file 4GB in size" (http://www.securityfocus.com/bid/3027/exploit/), to potentially cause a denial of service. This setting reduces that risk.

- Limit attachment + message size to 20480Kb, and number of attachments to 20. This allows large messages to get through if needed for legitimate business purposes, but helps limit those of size or number that could cause service problems.

- Limit cleaning attempts to 20 for messages with multiple infected attachments, and number of viruses reported to 20. This setting reduces the work for the virus filter in unlikely cases where messages contain extreme numbers of viruses.

- Limit attachment size to 10Mb. Like the total message limit, this helps prevent potential service issues caused by overly large attachments being sent to or from the organization.

Configuration→Update

- The 'Update Now' feature was used to update the pattern file, scan engine, and spam database prior to the systems going online. This was achieved most easily by stopping the Trend Micro InterScan Messaging Security Suite for SMTP service whilst leaving the other Trend Micro services running. By manipulating services this way, no e-mail would be processed, but the update services could gather the updates from Trend Micro's Internet update server via HTTP.

- Scheduled update settings were then set to automatically check (and update if available) the pattern file, scan engine, and spam database every hour. This key setting was enabled to reduce exposure to new virus threats, as updates can be in place within one hour of release from Trend Micro.

Configuration→Logs

- For log maintenance, log level was left at 'Normal'. Higher settings created enormous amounts of logs, which were found to be excessive under normal circumstances. In situations where problems were experienced and needed support assistance, log levels were set to 'Diagnostic', but only for long enough to replicate the issue. The number of days to keep logs was set to 60 days, with maximum size set to 512000Kb. These settings will depend on space availability on the server, and will change for each environment, but it is best to retain as much log information as possible on the server prior to archiving off to other locations, to assist any user support issues about e-mail problems.

Configuration→SMTP Routing→Receiver

- In the 'Settings' folder, the server IP address was simply set to the valid IP address of the system. A single-homed server is recommended, to remove any additional routing and other complications introduced by a dual-homed environment. The standard SMTP port was used (25), and the SMTP server's greeting message modified to remove any reference to the software used, i.e. 'SMTPGWY, ready at '. Whilst this doesn't completely prevent external users from determining SMTP software details (e.g. they could still examine message headers), the added measure does make it more difficult for would-be attackers, albeit only by adding 'security by obscurity'.

- Connections were set to timeout after 10 minutes of inactivity, and a limit of 1000 simultaneous connections enforced to prevent overload. Reverse DNS lookups were not enabled due to the performance impact

causing message delays, however would be recommended in an environment where sufficient server capacity and bandwidth are available.

- Connection control was configured to accept all addresses, because the server was Internet-facing.

- Relay control was implemented to only allow internal domains as destinations, and internal messaging systems as senders of relayed mail (i.e. to any destination). These settings are crucial in preventing relaying issues and potential blacklisting by open relay monitoring sites.

Configuration→SMTP Routing→Delivery

- Internal domains were added to the 'Domain-Based Delivery' section, and set to deliver via the internal messaging system. All other domains were configured to be delivered to via DNS – note that DNS servers can be included here, but the application will simply use the operating system settings if not, so it is best to avoid doubling up by not entering any.

- In the 'Advanced' section, the retry interval for delivery failures was set to 30 minutes, with a 24 hour maximum retry period. This is fully customizable, and in environments with low e-mail volumes, could be increased significantly. The option to disable insertion of the "Received:" header during message processing was not used, because although it can arguably increase security by hiding the SMTP gateway server in message trails, it is causes non-compliance with RFC 821:

  > When the receiver-SMTP accepts a message either for relaying or for final delivery it inserts at the beginning of the mail data a time stamp line. The time stamp line indicates the identity of the host that sent the message, and the identity of the host that received the message (and is inserting this time stamp), and the date and time the message was received. (Postel, p.21)

Configuration→SMTP Routing→Message

- All settings were enforced, with maximum message size 10240Kb, maximum data size 10240Kb, maximum messages 100, and maximum recipients 100. Note that the maximum messages per connection setting may present problems, depending on the behavior of internal and external senders. In our environment, the limit above was found to be a realistic one to allow valid senders and prevent rogue servers from tying up our servers.

Configuration→System Monitor

- The 'Event Monitoring' section was very useful for configuring alerts to be sent based on the thresholds of 2000 for the delivery queue, 5

minutes for the SMTP service stopping, and 400Mb for processing queue free space. The alert messages were updated to include the threshold values, so that if and when alerts were received, problems could be identified and resolved quickly due to the meaningful messages. SNMP trap notification was enabled in addition to e-mail, so that internal monitoring systems could automatically page appropriate support staff when needed (particularly after hours). Enabling alerts for the scheduled update result also allowed for a further system health check, as this occurred on an almost daily basis.

The 'Policy Manager' configuration area allowed further management of alerts and message content rules. Many policies, such as attachment blocking, are included natively within IMSS, and only require fine-tuning to suit a given environment. In our case, the following key policies were implemented:

Policy Manager→Filter Action

- It was found to be best to setup filter actions prior to assigning them to policies, to save time when configuring policies. Filter actions were added using meaningful names, such as 'Delete and Notify Administrator' or 'Quarantine and Notify - Virus', based on their function. The setting to 'Attach modified message' to the alert, particularly to the Administrator, was found to be very helpful in resolving support queries from users, and is highly recommended. Another useful technique was to code the subject such that a quick look could allow the administrator or support staff to determine which server sent the alert (this applies to all alerts from the system to support staff).

Policy Manager→Quarantine Area

- Quarantine folders were setup for specific filters – virus, attachment blocks, and spam. This increased the ability to report on numbers of messages filtered over time, as well as message sizes. To avoid unnecessarily filling drive space, all areas were configured to delete messages older than 60 days.

Policy Manager→Global Policy

- The anti-virus policy was modified to ensure that all files were scanned, and to delete infected files that were not cleanable. A warning message was applied to adequately notify the sender and receiver, i.e. 'The attachment (%FILENAME%) was infected with a virus and removed.'. To properly test the anti-virus policy, the "eicar" test files (http://www.eicar.org/anti_virus_test_file.htm) proved to be very effective in verifying detection of standard attachments as well as attachments that were zipped in multiple layers. The test files are also safe, as opposed to using live virus samples.

- A disclaimer policy was added, utilizing the 'Disclaimer Manager Filter' group, but only applied in an outgoing direction.

- An 'Attachment Removal' policy was added, utilizing the 'Message Attachment Filter' group, and initially configured to automatically strip attachments with the following extensions:

  *.exe, *.scr, *.bat, *.shs, *.vbs, *.pif, *.dot, *.pot, *.com, *.cmd, *.lnk, *.eml, *.swf, *.vbe, *.chm

  The above was implemented because it allowed proactive removal of unwanted and potentially dangerous message content, to further reduce the risk introduced by new viruses before updated pattern files became available. It was largely based on the "Level 1 ("Unsafe")" attachments list that was introduced by Microsoft with the Outlook security patch, which is defined on the company web site as "any extension that may have script or code associated with it" (http://support.microsoft.com/?kbid=290497). Not all were allowed to be automatically stripped, due to the needs of internal and external customers (i.e. the security risk alleviated by blocking some files was outweighed by the increased cost to business), but the more dangerous and commonly used ones, i.e. by mass-mailing viruses/worms, were included for the most part. It is also important to note that the list above is a living one, and can be adjusted according to new vulnerabilities, their level of exploitation, and resulting risk to the business.

- The anti-spam filter was also enabled, and set to use the automatically updated spam rules.

- Filter order was adjusted to ensure that each message was processed first against the attachment blocking filter, thus making the scanner more efficient and not overloading the virus filter unnecessarily. In environments where message numbers are small, and monitoring resources large, the virus filter could be placed first to provide more accurate feedback as to the number of viruses detected.

Policy Manager→Global Policy→Incoming Policy

- Routes were configured to determine incoming direction for messages, by adding each domain that would be used as a 'From:' address for our organization.

Policy Manager→Global Policy→Outgoing Policy

- Routes were configured to determine outgoing direction for messages, by adding each domain that would be used as a 'To:' address for our organization.

Additional configuration involved fine-tuning some settings in the IsntSmtp.ini file, found in the root of the IMSS installation folder. To enable IMSS to better utilize available hardware resources, the following values in the [EMail-Scan] section were increased from defaults, based on feedback from Trend Micro Support (an example can be found at the following Knowledge Base article: http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=13233):

- ScanningThread – increased from 10 to 50
- PickupDeliverThread – increased from 3 to 10
- PickupScanThread – increased from 3 to 10
- MailQueueThread – increased from 20 to 40
- BounceMailQueueThread – increased from 5 to 10

The above changes produced a dramatic change in performance, increasing the messaging processing capacity and reducing any delays significantly. It would be recommended to change these values as hardware permits, by adjusting the values incrementally, then restarting the SMTP service and monitoring server performance.

Because the solution was able to leverage off existing hardware, cost to the business was significantly reduced in comparison with a solution that would have required the purchase of additional appliances or other hardware with increased capacity. Costs were further reduced by being able to utilize the existing and internally supported operating system baseline (Windows 2000 Server), meaning that additional training was only required for the IMSS application itself.

Security was further enhanced by limiting connectivity for the IMSS server via the internal and external DMZ firewalls, i.e. SMTP (TCP 25) between it and the Internet and internal messaging systems, DNS lookups (UDP port 53) to the Internet, and HTTP (TCP port 80) to the Internet to allow updates to be automatically downloaded.

After

Problem Resolution

Implementation of IMSS into the SMTP gateway resolved very effectively the lack of virus and message content filtering, adding a robust perimeter anti-virus layer to the network. Since its introduction, viruses and worms such as 'Bugbear' variants, that were very successful in infecting many organizations around the globe, have not presented a problem to our network. Apart from a slightly increased load on IMSS due to the sheer message numbers arriving from external sources, these outbreaks were handled without any problems becoming evident. Message numbers of up to 50,000 per day have caused no performance problems, with several thousand viruses and attachments successfully blocked, as well as 1,000-2,000 spam messages quarantined daily. Although much spam still reaches internal users, the automatically updated rules to filter the majority of cases, and to a great extent prevent users being overwhelmed by the problem.

Whilst the ability to update automatically reduced the implementation time for new virus outbreaks to be detected, as installation occurred within an hour of pattern file release by Trend Micro technicians, the attachment blocking policy proved to be most effective in preventing new viruses from penetrating the internal network. Due to our large message volumes, and number of external customers with Internet connectivity, we would on occasion notice many samples of new viruses (usually obvious due to patterns in the message headers or attachment name) being filtered by the attachment blocking policy several hours before detection was available via the virus policy due to updated pattern files.

The solution also provided an increased incident response capability, with the ability to quickly identify outbreaks externally, and if necessary escalate internal protection measures.

Additional Complications

Several complications were encountered during and after the implementation, although they were resolved in most cases by patches made available by Trend Micro support. Examples are a memory leak which was quickly detected by monitoring of the server, and the message processing halting due to specific message content causing the application to fail to pass messages from the processing queue to the delivery queue. Restarting the Isntsmtp.exe process resolved most issues temporarily, if a fix was not available, although if the server was under load, this was best accomplished by either rebooting it or using the Resource Kit tool KILL.EXE. The CriticalError.log file produced by IMSS also proved useful for tracking messages that were causing problems for the SMTP service. Once Service Pack 1 was applied, no serious problems were observed, with the system operating very smoothly. The configured alerting also

aided in reducing any downtime, as issues were discovered very quickly as queues increased. Setting the appropriate thresholds for the alerts was largely a matter of trial and error, weighing up expected load in conjunction with known problem periods.

Potential Vulnerability and Risk Issues

Whilst many vulnerabilities were reduced or removed by the IMSS implementation, there are still potential weaknesses in the new solution. Even though several message attachment types are proactively filtered, some are still allowed through that could contain malicious content. An example is a password-protected compressed attachment, like a .zip file created using WinZip, that may contain many other files. The scanner will detect known malicious code samples within compressed attachments to a degree, but not if password-protected. Attachments of this nature will be allowed to bypass the SMTP gateway filter, and reach the end user. From there, the execution of code won't be automatic, but is a possibility (note that users have been known to seemingly try very hard to bypass screening systems if they think the message content is worth their friends seeing, which is why the likes of Happy99 was so successful – users thought it was a legitimate file, because it "shows a fireworks display to disguise its installation" (http://www.symantec.com/avcenter/venc/data/happy99.worm.html) and so ignored warnings and found ways to send it without triggering messaging anti-virus software.

Additional risk of service interruption remains due to the implementation not being clustered or load-balanced. Although the internal messaging systems will queue messages if the SMTP gateway is not reachable, there is a potential for delays to occur because a manual change is required to redirect traffic to the contingency server. Whilst from the Internet, MX records effectively allow redirection, a limitation with the outbound connection means that unexpected system problems will cause issues that may be noticed by users.

Automated updates, whilst very useful for reducing response time for new virus outbreaks as well as manual support staff operational tasks, also introduced the possibility that a corrupted pattern or spam-rule file could cause service disruption. This was deemed to be a very low risk, compared with the benefits gained by automated updates, and because any problem introduced by a corrupt update would be likely to be resolved very quickly as a high priority by Trend Micro support.

A lack of reporting also presents a potential risk, given that it is difficult to provide support staff or the business with accurate numbers of viruses and other items filtered, without setting up third party log parsing utilities. Although the alerting mechanisms are very effective, without long term reporting capabilities it becomes difficult to identify trends that could eventually cause service problems if not taken into account early.

Complications Introduced

Other complications introduced by the advanced filtering at the SMTP gateway, were due to the increased support requirements that resulted. Prior to IMSS, the internal messaging system had to cope with all the large messages and viruses, but afterwards most of the issues raised by users who had not received messages they were expecting were deemed to be (and usually were) due to the gateway.

Conclusion

The risks inherent in not having an SMTP gateway virus filter, as identified in conjunction with the SANS Security Essentials coverage of Defense In Depth strategies, were significantly reduced by the implementation of IMSS. New problems and risks introduced were very much outweighed by those mitigated by its use.

Further enhancements could be achieved through better reporting functionality for support as well as the business, and through load balancing or clustering multiple servers to reduce potential downtime and increase service capacity.

Whilst virus filtering is typically a reactive measure in terms of detection and cleaning capabilities, IMSS policies have allowed messages to be more proactively filtered for attachments and other content that were unwanted, and that could easily have caused many security problems and support nightmares if allowed to pass undetected.

List of References

Bridwell, Larry. “ICSA Labs 8th Annual Virus Prevalence Survey.” http://www.icsalabs.com/2002avpsurvey/ (5 Jul. 2003)

“Trend Micro Virus Information, virus alerts, advisories, Top 10, antivirus, worm, trojan, macro.” http://www.trendmicro.com/vinfo/default.asp?sect=TT (6 Jul. 2003)

“WORM_LOVGATE.F – Description and solution.” http://www.trendmicro.com/vinfo/virusencyclo/default5.asp?VName=WORM_LOVGATE.F (6 Jul. 2003)

“Top ten viruses reported to Sophos in June 2003.” http://www.sophos.com/virusinfo/topten/ (6 Jul. 2003)

“Solution 13668.” http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=13668 (6 Jul. 2003)

“SecurityFocus home vulns exploit: Multiple Vendor File Scanner Malicious Archive DoS Vulnerability.” http://www.securityfocus.com/bid/3027/exploit/ (6 Jul. 2003)

Postel, Jonathan B. “RFC 821 SIMPLE MAIL TRANSFER PROTOCOL.” http://www.ietf.org/rfc/rfc0821.txt (6 Jul. 2003)

“eicar – Anti-Virus test file.” http://www.eicar.org/anti_virus_test_file.htm (6 Jul. 2003)

“290497 – OL2002: You Cannot Open Attachments.” http://support.microsoft.com/?kbid=290497 (6 Jul. 2003)

“Solution 13233.” http://kb.trendmicro.com/solutions/solutionDetail.asp?solutionID=13233 (6 Jul. 2003)

“Symantec Security Response – Happy99.Worm.” http://www.symantec.com/avcenter/venc/data/happy99.worm.html (6 Jul. 2003)