

PARADIGM SHIFT – FROM STATIC TO REALTIME, A PROGRESS REPORT

*Matt Garrad, Paul Jones, Lysa Myers, Michael Parsons
West Coast Labs*

About Authors

Matt Garrad, Paul Jones, Lysa Myers, Michael Parsons

Lysa Myers is the Director of Research for West Coast Labs, a leading independent test facility for information security and threat trends. She is responsible for researching and analyzing IT threat trends, reviewing and developing test methodologies. Myers spent 10 years in the Avert group at McAfee Security, Inc. coordinating researchers to create detection and removal solutions, and training security researchers both within McAfee and at US government agencies. She is a member of numerous security industry groups, including the Drone Armies mailing list. A highly sought after expert resource on security topics, Myers is a regular conference presenter, and an IANS faculty member.

Michael Parsons was born in Cardiff, Wales, in 1959. He read law at Cambridge and joined the Civil Service, moving in 1986 into a support role of the mainframes at the Driver and Vehicle Licensing Agency, where he specialised in supporting the security administration, playing a part in their conversion to generic protection. In 1995 he joined West Coast Publishing Ltd., (later West Coast Labs), supporting the computer equipment and carrying out testing and reviews. In 1996 he became the tester for the new Checkmark certification, eventually becoming the Content Security Labs Manager. In his current role as Senior Malware Researcher for West Coast Labs he administers their Honeynet and Test Suites. He has appeared on a panel at VB and his hobbies include reading and collecting books (over 9000 at the last count), going murdering at the weekends about three times a year, and correcting other people's mistakes (E&OE).

Paul Jones is a Test Engineer at West Coast Labs and is based in the Cardiff, Wales office. A Welsh native, he joined the organisation in 2006 and spent the first month conducting research so secret he couldn't even tell his mother about it. His professional interests include programming, virtualisation, and test automation. He is heavily involved in both the Honeynet and Real Time systems, providing much of the

proprietary code used in testing. He is currently decorating his new house, and hopes to make it habitable soon.

Matt Garrad is Director of Technical Services at West Coast Labs and is based in Cardiff, Wales. His professional interests include network and content security, programming (including legacy languages), databases and automation, data visualisation techniques, and web technologies. Prior to joining the organisation in 2005, he worked in a series of jobs including spells in a UK University as an Oracle/web programmer and an Oracle DBA. When not working, he enjoys spending time with his wife and young son and is a guitarist / vocalist (but not singer) for a thrash metal band. He remains proud of the fact that he is one of the few people in the world to get thrash metal played on BBC Radio 4 (twice!), even in the face of his wife's continuing embarrassment.

West Coast Labs, Unit 9 Oak Tree Court, Mulberry Drive, Cardiff Gate Business Park, Cardiff, CF23 8RS UK

Telephone : +44 (0)29 2054 8400

Facsimile : +44 (0)29 2054 8401

Email: {mgarrad, pjones, lmyers, mparsons} @westcoast.com

Keywords

Static Testing, Dynamic Testing, Testing Paradigm Shift, Real Time Testing, Attack Vectors, Case Studies, Time To Detect, Sample Set, Global Malware Trends, HoneyPots, HoneyClients

PARADIGM SHIFT – FROM STATIC TO REALTIME, A PROGRESS REPORT

Abstract

There has been much discussion, in the past couple of years particularly, regarding the best way to go about testing anti-malware products. Being a testing organization, this is a subject to which West Coast Labs has given considerable thought. There have been many changes to the existing testing setups in the efforts to shift the testing paradigms. This paper looks to discuss the many things that have been investigated and give a view into some of the results this has turned up.

In order to bring testing back in line with user experience, it is necessary to change the timing and nature of the tests themselves. Not only does this mean a change in overall methodology of processing the samples, but a change in how results are analysed as well. This has also required a new model of gathering and verification of samples to ensure a fresh and timely sample set, which has presented its own set of issues. Once all that is done, it is then important to find a meaningful way to present the data to users. As this is a process which does and should continue to change indefinitely, this paper presents the most up-to-date report of West Coast Labs' progress.

Introduction

Once upon a time, there were but a handful of viruses. Anti-virus updates came periodically, because viruses were released infrequently enough that once-a-month updates were entirely sufficient. Users could scan their machine once a day, “on-demand”, and be sufficiently protected. It was in these early days that the anti-virus testing industry was created. It was sufficient to put products through their paces a few times a year, against the entire universe of threats which might be liable to infect a user's machine. These static tests mimicked what a user was likely to experience.

Obviously, this is no longer the case. (Cue portentous organ music)

In the last few years, there have been a number of significant changes in the testing industry, to better bring it in line with current products and user experience. The previous, static variety of test still has value as a benchmark of a minimum level of performance which products must meet. But to completely evaluate a product, much more rigorous testing is needed.

What are the main kinds of tests?

Static testing is the oldest variety of testing. Usually, the first step would involve getting a collection of malicious and known-clean samples to test. Originally, the malicious files were from the Wildlist⁽¹⁾, but lately various testing agencies have preferred to use their own test-set, either in addition to or instead of the Wildlist. These newer test-sets can vary widely in number, and it's a matter of competition to get the largest or most relevant sample set – often diametrically opposed goals. The next step involves obtaining a product, showing the test files to the solution and observing if the product detects them without alerting on known-clean files. The results should be reproducible, and this is usually ensured by completely documenting the steps taken. Once the test

is complete, the tester takes their documentation and the results and gives them to the product vendor, to verify that the results are accurate. This test is obviously quite simple, but it gives both customers and product vendors a way to ensure a basic level of acceptable protection and performance.

The more advanced varieties of testing include retrospective testing, real-time testing, and dynamic testing. Retrospective testing was the first kind of testing after static testing to be introduced. It is an evolved form of static testing where older virus definitions are used against brand-new samples, in order to specifically test heuristic and generic detection capabilities. Evaluating generic detection, one still tests signature-based detection, but a more advanced type of signature covering groups of code rather than an individual program, while heuristics study the behaviour of files rather than the code used to create such behaviour - and so need yet another type of evaluation. In these tests, having a good set of known-clean files is especially important, as these advanced signatures and heuristics are usually more prone to false-positives.

Dynamic testing differs primarily because test files are actually executed. This approach tests not only signature-based detection but also run-time detection. Many products are now sold as suites which include technologies such as basic Host-based Intrusion Prevention Systems, and even more products contain additional anti-malware technology such as behaviour-based scanning. This type of testing is considerably more time-consuming, as each individual file must be examined and allowed to execute, with the results subsequently analysed on a sample by sample basis. Sample sets are generally significantly smaller, so testing organizations should take care to ensure that the most relevant samples are included.

Real Time testing will be the focus of the majority of this paper, as the majority of West Coast Labs' efforts have focused on this type of testing for the previous two years. Real Time testing is continuous, all day every day. Rather than doing tests once every month or quarterly, using sample sets which are solidified before each test period, samples are gathered and tested constantly. Files which are undetected can be repeatably and repeatedly sent back through each product to determine how long it takes to add malicious files to detection. As vendors add new malware to their detection capability on an ongoing basis, this approach more accurately tests, mirrors and verifies a vendor's research and protection efforts. This sort of testing, as it is ongoing, can never be a "pass/fail" as in a traditional test. It is only possible to report a percentage of detection, which will continually fluctuate, much like a stock index.

Of course, it is possible to perform other kinds of tests which focus on different facets of anti-malware detection technology (as well as things like performance testing) and each has its own unique set of difficulties. Models looking at malicious URL testing and cloud-based testing are most notable among these, as it requires yet another paradigm shift on the part of the tester, to one which does not have "reproducible" results due to the extremely volatile and temporal nature of the content of the URLs and cloud-based virus-definitions. In these cases, it must suffice for the testers to record their actions and enough data to be able to satisfy the requirements of any later inspection in order for the vendor to verify that the tester's actions were correct at the particular point in time.

Creating a Real Time network

One of the most significant tasks in the past 2 years at West Coast Labs has been the development of the Real Time test network. As with any significant change, this has been a monumental undertaking full of interesting twists and turns. The main tasks after deciding the specifics of the testing methodology were to gather samples, create the implementation of the methodology, create a secure interface for vendors, and to decide what to do with the data that was being generated.

Sample Selection and gathering

The first task in this endeavour was to gather samples. As the test is continuous and ongoing, it is possible to include an increasingly large number of samples as they're received and processed over a longer period of time. The primary goal is to obtain samples in order to continue to mirror what is likely to infect customers, which means that there is a need for samples from a variety of different attack vectors, from all over the world. West Coast Labs is, in this case, fortunate in that Haymarket Media Group (West Coast Labs' parent company) is a global publisher, and it is possible to put sample collectors in offices on almost every continent. Initially the focus was only on a few basic attack vectors, but ongoing work is based around the continual updating and addition of collection methods to include more attack vectors, to continually represent and provide coverage for the technologies added and monitored by the malware industries.

As the decision had been made to recreate the environment of a small to medium business, the initially focus was on the threats people are likely to face on a machine "out of the box" – the threats which will hit an internet-connected computer before a user even begins to start doing the standard "user tasks" such as checking emails or surfing the web. These are primarily network-aware worms, which spread quite well with zero user-interaction. The collectors that used are entirely unprotected by any AV, and so it is possible to see a true picture of what is "in-the-wild" as opposed to seeing what is in the wild and doesn't get stopped by a patched AV solution - therefore a significant number of these threats are several years old, including in some cases malware that is almost 15 years old, yet is still spreading via non-protected and non-patched Windows machines.

While some examples are delivered in short, sharp epidemics before disappearing, others continue to flood in. Each week West Coast Labs assemble a prevalence table, showing the 30 files most frequently delivered to the HoneyPot network that week. Of the 31 (a tie for 30th place) in the most recent week's table at time of compilation of this report, 14 had appeared in the same table 3 months before, 17 in the same table 6 months before and 13 in the same table 9 months before. However, the majority of the malware seen are brand-new (at least to West Coast Labs' collections) and potentially undetected variants of established malware families.

Collection of old malware may strike some people as a little pointless and unusual, but there are two very good reasons for doing this. Firstly, it is obviously still in the wild and still spreading around machines that are unprotected by an anti-malware solution, are protected but the updates for the

solution are out of date, or have unlicensed copies of the operating system, or possibly some combination of the above. Secondly, given that it is possible to observe instances of some companies failing to detect these older samples, it shows that whilst there is a great need and push within the industry to catch the latest and greatest malware, some of the older pieces of malware are still being undetected or are dropping out of detection – a problem that has been ongoing for years⁽²⁾. It is not the place of the authors to speculate on whether this is due to signatures and updates being rotated out for space considerations or whether the companies concerned have just not seen these pieces of malware before, but either way it is a potentially worrying situation.

No one will say that these threats comprise a large percentage of threats on the internet, so more attack vectors need to be included. Email threats were the obvious next place to go, as this has been a popular attack vector for a number of years and it continues to be so – indeed MessageLabs⁽³⁾ reported an increase in viruses to 1 in every 302.8 messages received in February 2010. West Coast Labs are in a fortunate position once again in that there are a number of sources of live malware via SMTP that can be utilized including independent feeds, feeds from the wider Haymarket Media group, and industry and commercial partners who provide us with samples and data.

After that, P2P has been considered somewhat of a cesspool for viruses for many years. Searching for viruses on this medium has proven to be a bit like shooting fish in a barrel. Malware is relatively easy to find, with a few basic search terms, and the samples found here have little overlap with other attack vectors, making it a valuable addition to the collection efforts.

The biggest percentage of threats right now is web-based threats: According to Webroot, in 2008 they comprised 85% of malware⁽⁴⁾. In 2009, Websense indicated that there was an above 600% rise in the number of malicious sites during Q1 and Q2⁽⁵⁾. But web-based malware is far from an easy thing to gather. With more traditional malware, it suffices to have a source of email and a bank of HoneyPots, (plus some means of copying files to a remote, secure location) then one can just sit and wait for the malware to roll in. Web-based malware requires a more active approach, going to where the malware is being offered. For this, the implementation chosen was to set up spidering services and URL collection methods that are linked to HoneyClient machines. This process then scours the web looking for malicious behaviour and possible points of infection. All URLs discovered are passed over to the HoneyClient processes, and once the machine contained therein has been shown to be breached, the output is fed into a stream of known bad links coming into the central test system. From this point on, the procedure is much the same as for other collection methods.

In order to ensure an ongoing supply of samples, each collector is checked regularly every 10 minutes, with new samples pulled in at that frequency. These are then pooled every hour, checked for the inevitable duplicates that occur, damaged and corrupted samples, and are then tested against a number of different solutions including both desktop and gateway. This approach ensures that the maximum time between a sample being seen for the first time in the remote collectors and that same sample being tested against the product should be just over an hour, and is often less than this.

This approach allows the leverage of the overabundance of samples already available and hitting actual customer machines to allow a focus on the day-to-day customer experience rather than attempt to reach the edge of “the infinite space” of malware or product potential. Importantly, samples in the Real Time systems are not created or modified in any way; the aim is to take a representative sample of what is already floating around cyberspace and test it as is. It is not the intention of the authors to either argue the validity of other approaches, or to discuss in-depth the specifics of weeding out extraneous samples as those are each discussions to be had separately to this paper. In short, there is no one sample set which should be considered to be comprehensive for all purposes. A sample set should reflect the aims of the test itself, to best illustrate the question being asked by the test.

Also hugely important is the ongoing two-way communication that is put in place with all the vendors involved in the testing that allows suspicious samples to be flagged. Upon receipt of a request to examine a file, each file is removed for further manual analysis before either being discarded or reintroduced depending upon the outcomes of these external investigations. The Portable Executable format corruption checkers employed within the system ensure that these are mostly kept to a minimum, with less than 1% of samples that have run through the system to date having been questioned by vendors.

Any samples missed by the solutions are made available to the vendors for download, and can (should the vendor wish, and have the capability to support) be streamed immediately to their backend servers for processing. This model allows for an almost immediate testing, feedback and data-gathering of threats.

Creating the Real Time Infrastructure

Code for the Real Time testing system is entirely proprietary and written in-house. Discussion of exactly how the code works and the processes involved in the collection, analysis, and distribution of the samples that are received are in-depth enough to almost require full separate papers on each method and are currently covered by our Commercial In Confidence rating, however a high level overview and application of the same standard across the numerous attack vectors is undertaken. This ensures that, as stated above, wherever possible the tests should be repeatable and repeated until a vendor detects the sample, and where this is not possible (due for example to temporal constraints), enough data should be recorded to substantiate any later presentation of results.

All vendors’ products are connected permanently with full access out to the internet to apply updates and are set to check for and download updates (where possible) on an at least hourly basis to ensure that signatures and updates are as fresh as possible. All are installed with default options unless specifically requested by the vendor, and where changes are made these are noted by the test team so that the conditions can be recreated.

Tests are sorted and broken down by the attack vector from which the sample originally came - for example, those malware samples collected over email can be replayed over SMTP or POP3 depending upon the acceptance configuration of the solution under test. Non-detected samples are resubmitted through the system every hour over the appropriate protocol until they are marked as detected.

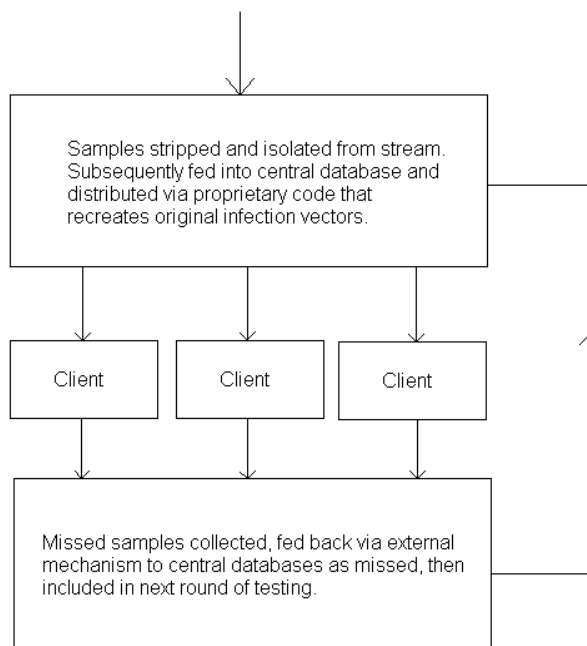


Figure 1: High level overview of re-feed mechanism (protocol independent)

Figure 1 shows a high level overview of the system with samples entering the system stripped from their original context and isolated from contextual packaging that may influence the outcome of any feed prior to testing, then fed through multiple clients simultaneously. All results are collected on the far side of each of the clients at a central server, analysed for any misses and then re-fed back into the central database along with the date and time of each test so that it is relatively easy to extract a list of Time to Detects (TTDs) on a per sample and per vendor basis.

Presentation of the results to vendors

Presentation of scanning results is performed by way of a secure online interface locked down using several different approaches so that vendors can only see their own results and these results are not available to the wider internet community. Data presented to vendors is represented as percentages for ease of interpretation, although actual figures can be made available should the vendor require it. The web interface is updated on an approximately 5-minute basis, depending upon the amount of processing ongoing on each feed at any given time point.

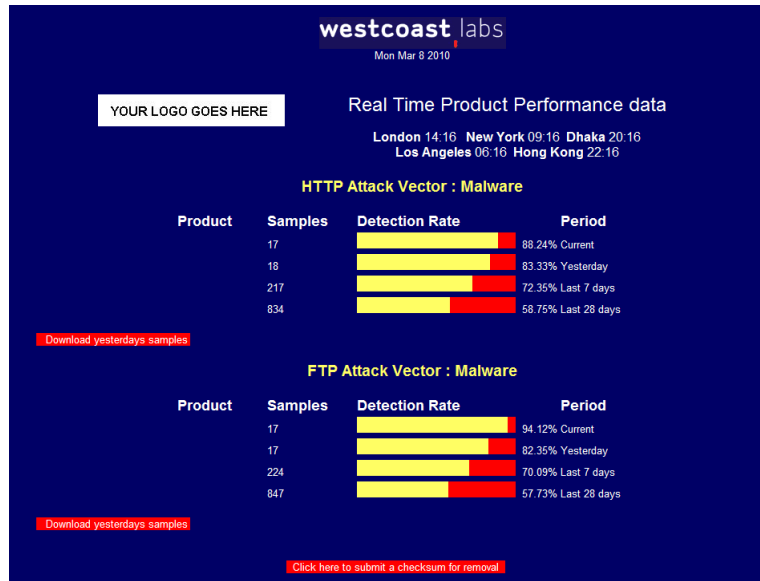


Figure 2: Example (anonymised) screen grab of the interface.

Figure 2 shows an example of the sort of interface that the vendors might see – with percentages detected marked in yellow, and those undetected marked in red. There are link buttons on the left to download the missed samples for the previous 24 hours, which cumulatively include all samples that are currently being missed (so therefore may include samples from several previous days if the vendors have not logged in), and a link button at the bottom for vendors to submit samples that they believe to be corrupted, clean, or otherwise incorrectly included in the feeds.

Currently data is not made available to the general public, although there are plans to provide some form of data - discussions are ongoing at the moment regarding the best way to present this in order to make it both relevant and accurate.

The presentation of the data is shown over several different time frames, enabling a vendor to keep track of how their product or solution is doing during the time frames chosen – in this case “Current” which reflects the 24 hour period that is ongoing, “Yesterday” (the previous 24 hours), last 7 days, and last 28 days. It is interesting to note that vendors have reacted well to this form of presentation, in that it gives them quick and easy access to the data and shows a progression (or regression!) over the last 4 weeks, thus feeding into their R & D programs.

The use of the interface has been taken up by both technical staff and project/product managers, as well as in some cases being available to the highest authorities in the company – who are justifiably concerned if their detection drops below what is considered an acceptable level. This acceptable level is generally set within the companies themselves as, although everyone would love to offer 100% detection (especially marketing managers!), the acceptance within the technical community that “protecting against unknown threats” is a fallacy is becoming widely accepted, and many of the vendors involved merely look for a high level of detection rather than 100%.

A surfeit of data and some high level interpretations

Naturally, this data is not being gathered for our health. An important question to answer is what does any of this actually mean to anyone, and is there any practical application for the results? The specifics actually paint a rather interesting picture of both product functionality and the geographically diverse nature of malware. Of course, any company grabbing huge amounts of data such as this has to both be careful that the data is of some use, rather than just being gathered for its own sake, and also ensure that any representations made using that data are fair and balanced. Later in this paper there are a couple of interesting case studies to show what is possible at a high level with the data that is being received.

Breaking down samples by attack vector allows the gathering of metrics related to how long it takes a vendor to add a file to detection, as well as any differences in the products' ability to protect against samples which come over more than one network protocol. For instance, a product may protect against particular samples over HTTP, but not protect against the same samples on FTP, and there are numerous examples of this occurring. Where samples have been observed being delivered over more than one protocol, it is important to have tested that one sample against each protocol that it is received on.

This has shown up some inconsistencies, notably where vendors have decided for space or efficiency reasons to limit the scanning during transmission of files over particular protocols to a specific file size limit. Upon further investigation it was shown with the vendors concerned that, when they were tested against the malware actually executing subsequent to the download, then the downloaded files were stopped from running, correctly identified, and the machines protected. This has had the benefit of leading some vendors to re-evaluate their limitations on such transfers and roll out alterations to their products to the wider community of end users.

It has also been observed that not only are there some global pandemics of particular virus families, but there are region-specific outbreaks that do not spread outside of (for example) the Far East, or in some cases even particular countries. Also of note is the verification that there are still some seriously old viruses floating around - a reflection on the fact that people still seem to use Word 97 or Windows 98 with no AV installed. As an example of this, within the two weeks prior to the submission date of this paper, multiple copies of a file identified as W97M/Thus.M turned up in the SMTP Malware feed.

Results related to malware attacking

West Coast Labs have extracted the following examples of data from the successful attacks (i.e. those that produce malware) related to countries and locations attacking our HoneyPots.

The top ten countries which have sent most unique pieces of malware are represented in proportion as shown in figure 3.

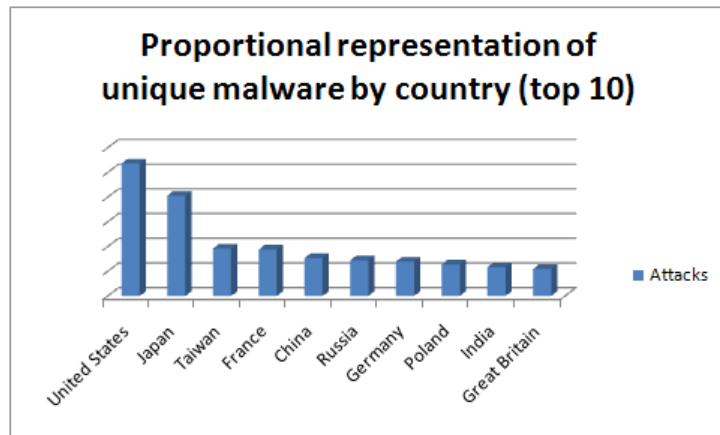


Figure 3: Countries sending most unique pieces of malware against the WCL HoneyPots

This can be compared with the following, which shows proportionally the number of attacks that result in pieces of malware (non unique), as shown in figure 4

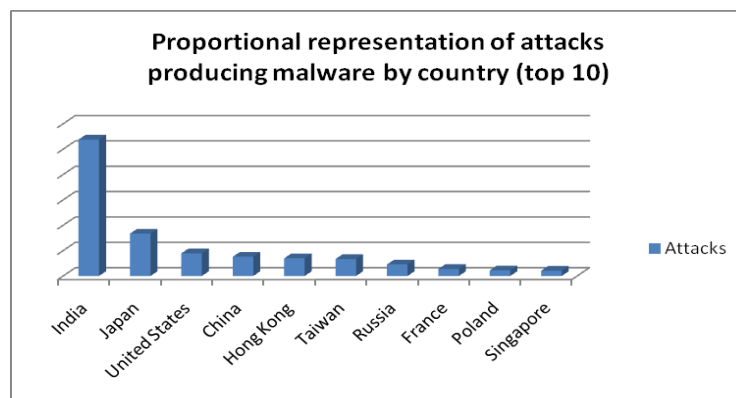


Figure 4: Countries most attacking WCL HoneyPots

It can be seen from here that, although there are a number of countries which fall into both top-10 categories, a large number of attacks does not necessarily result in a large number of unique pieces of malware.

Figure 5 shows a wider global overview of the attacks that have been sourced by region, although these are tempered by knowledge that a reasonable proportion of the attacks originating from South and Southeast Asia never left that region.

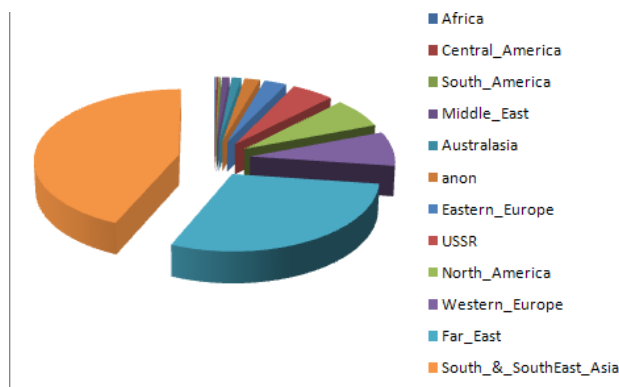


Figure 5: Global overview of attacking zones against WCL's HoneyPots

The ten individual IP addresses that have been seen to be producing the greatest number of unique pieces of malware resolve back as follows – 3 each are in China and India, two are in the Republic of Korea with one each in Egypt and Vietnam. Interestingly, the number of IP addresses making only 1 unique infection attempt (ie one piece of malware delivered either once only or on multiple occasions) makes up 86.96% (rounded up to two decimal places) of all malware-producing attacks against the network.

A further 8.92% (rounded again) have produced two unique infection attempts, and as should be expected the proportion of the overall total diminishes with the increasing number of unique pieces of malware, to the point where it can be seen that individual IP addresses that are producing 10 or more pieces of malware make up only 0.14% of the attacks that have been observed against our HoneyPots.

Also, as an aside, when considering time frames, those IP addresses that have attacked only in one 24 hour period make up 91.07% of the total number of attacks, with those that have been consistently attacking for more than a year making up just under 0.1% of the total.

A case study of one attacker

The longer term and higher sample providing IP addresses are potentially where the interesting stories lie, so to follow up on this, one IP address which produced 15 different pieces of malware that attacked our HoneyPots over a 4 month period during 2009 has been tracked back.

The first interesting point is that the IP address we chose to trace was based in Japan, second in our figures both for proportionally producing both the most unique pieces of malware and the most attacks that produced malware. Running a series of traces back, we discovered that the IP address was registered to a large global security company (who shall remain nameless) who provide services related to several aspects of security – from physical to electronic. Their Japanese base

covered some of the electronic protection that their company offered including IP-based CCTV systems, Biometric Systems, Access control and some airport security systems. The particular IP address that was attacking our HoneyPots turned out to contain a web server that controlled access to several of their customers' CCTV systems and the infections included several pieces of malware identified as Backdoors. Whether this subsequently has led to any control of this machine or the site on it being ceded, or whether any customer data was being leaked is beyond the remit of this investigation, but should perhaps be adjudged a cause for concern nonetheless.

Looking at the individual attacks, there were 109 distinct attacks that produced a total of 15 pieces of malware, some delivered once only, and others delivered 15, 17 or 22 times each. All attacks have been made against one individual location in the HoneyPot network.

On the first day that this IP address attacked (17 January 2009), 37 attacks were made using 7 different pieces of malware. There was then an 11 day gap before the next attack, when 3 pieces of malware reoccurred with 1 new piece, 20 attacks in all. On 5 of the 6 following days, a total of 23 attacks were made, reusing 2 of the 8 pieces of malware and introducing 2 new pieces.

15 days later, there were 23 attacks over 3 days (18 – 23 February 2009). On 4 March 2009, there was 1 attack of a new piece of malware and on 4 more dates in April and May there were 23 more attacks featuring that same piece of malware and 2 new pieces. We have had nothing since 8 May 2009. The biggest time gap between first appearance and last appearance of any piece of malware (that was observed by West Coast Labs) at this particular location was 18 days.

Two of the pieces of malware attacking were appearing in the contemporary Wildlists. W32Kolabc!ITW9 and W32Kolabc!ITW10 both arrived in January – the former joined the Wildlist in the January and the latter in February.

Of further interest when looking at the wider picture is that 9 pieces of malware in the worldwide and deduped HoneyPot collections were seen only from this attacker, with 3 others attacking only this same HoneyPot from other IP addresses, and the remaining 3 also seen in other HoneyPots.

Results related to Time to Detect

Of course, a principal piece of data that will interest both vendors and end-users alike is TTD, i.e. the amount of time between a solution or company seeing a piece of malware on the feed that they miss, and them subsequently adding it to their databases. Once again, there are a few examples of this, but a single case study may prove beneficial in showing the sort of data that it is possible to produce.

Examination of one particular sample is undertaken here because it tells an interesting story in terms of individual vendors' TTDs. Also, this file is the second most prevalent file that the

HoneyPots have seen during the first two months of 2010, accounting for 9.07% of the total attacks on the HoneyPot global network and having attacked almost half of the locations where collectors are placed.

In order to illustrate the point, a random cross sample of 7 vendors from those connected to the system at the time is included here to represent the type of data that is being collected.

The sample in question was first introduced on 2nd January 2010 to the test system, with it being detected by 3 of those vendors within the test/retest period of one hour, so to all extents and purposes immediately. The remaining 4 vendors had varying detection times from 25 hours up to 233 hours in one case. This sample has been identified as a member of the Buzus family, which is well known and widespread.

Conclusions

This leads to the question of what can be drawn from this data. Admittedly, this is high level results presentation, and this paper includes specific examples extracted to illustrate the point but it would seem to lead to the interpretation that there is a raft of interesting data that is being collected that can be put forward for further analysis.

It is possible to show that, in several cases and for specific examples, not all vendors are receiving the same samples at the same time independently of the provision of samples via the Real Time system, and that not all vendors are introducing signatures quickly into their databases. It is easy to understand that vendors can get hundreds of thousands of samples a day ⁽⁶⁾, and so perhaps in future calculations of effectiveness, this should be factored in, but to the end user, the number of samples a vendor gets each day is immaterial – all they are concerned with is the age old question “Am I protected?”. This data would go to show that there is no one good answer to that. For example, the company that took 233 hours to add the sample mentioned in the example above can also be shown to have immediately identified other samples which have taken the other companies here several days to add.

Also of interest is that the majority of attackers that we have seen appear to have only one infection on their machine that is being distributed – there are, however several factors that could go into this, DHCP handouts on non-business lines being just one example.

This data certainly shows that there is a significant amount of analysis that can be performed and West Coast Labs will be focusing on producing and presenting more granular data of this kind in the future, thus directing the efforts of the Research group. Such data is also of use to those vendors hoping to make their processes more efficient, as well as those with an interest in the global nature of infections and malware spreads.

References

- 1) <http://www.wildlist.org/>
- 2) Virus Bulletin Magazine, September 1998, pp. 18
- 3) http://www.messagelabs.co.uk/mlireport/MLI_2010_02_Feb_FINAL.pdf
- 4) http://www.webroot.com/En_US/about-press-room-press-releases-web-threats-more-pervasive-than-email-threats.html
- 5) http://www.websense.com/site/docs/whitepapers/en/WSL_Q1_Q2_2009_FNL.PDF
- 6) <http://blogs.technet.com/mmpc/archive/2009/04/30/protecting-our-customers-from-half-a-million-new-unique-malicious-files-every-day.aspx>