Analyzing the Accuracy and Time Costs of Web Application Security Scanners

By Larry Suto
Application Security Consultant
San Francisco
February, 2010

Abstract

This paper is intended as a follow-on study to my October 2007 study, "Analyzing the Effectiveness and Coverage of Web Application Security Scanners." This paper focuses on the accuracy and time needed to run, review and supplement the results of the web application scanners (Accunetix, Appscan by IBM, BurpSuitePro, Hailstorm by Cenzic, WebInspect by HP, NTOSpider by NT OBJECTives) as well as the Qualys managed scanning service.

In this study, both 'Point and Shoot' (PaS) as well as 'Trained' scans were performed for each scanner. In the 'Trained' scans, each tool was made aware of all the pages that it was supposed to test, mitigating the limitations of the crawler in the results. This was designed to address a criticism by some security professionals that PaS, the technique used in the 2007 study, is not an appropriate technique to scan web applications and that only manually trained scanning is appropriate.

The study centered around testing the effectiveness of seven web application scanners in the following 4 areas:

1. Number of verified vulnerability findings using **Point and Shoot (PaS)**
2. Number of verified vulnerability findings after the tool was **Trained** to find the links on the website
3. Accuracy of reported vulnerabilities
4. Amount of human time to ensure quality results

Given the large number of vulnerabilities missed by tools even when fully trained (49%) it is clear that accuracy should still be the primary focus of security teams looking to acquire a web application vulnerability assessment tool.

The results of this study are largely consistent with those in the October 2007 study. NTOSpider found over twice as many vulnerabilities as the average competitor having a 94% accuracy rating, with Hailstorm having the second best rating of 62%, but only after additional training. Appscan had the second best 'Point and Shoot' rating of 55% and the rest averaged 39%. It should be noted that training is time consuming and not really practical for sites beyond 50-100 links. As such, sites with a large delta between trained and untrained results (Accunetix, BurpSuitePro and Hailstorm) may require additional effort in large scans. One of the most

surprising results was the findings for market share leader WebInspect, which consistently landed at the bottom of the pack in its ability to crawl the sites and find vulnerabilities; it missed approximately half of the vulnerabilities on its own test site.

Introduction

When reviewing scanners, most vendors provide and host website(s) which are intentionally vulnerable in various ways. Web application security scanner vendors have seen a large number of vulnerabilities from varying web applications through their research and through their work with their clients. Vendors will often add newly discovered vulnerabilities to their test websites as they look to augment the capabilities of their scanner. As a result, these test applications represent the sum total of thousands of hours of research and thousands of real world scans and are a fairly good representation of the types of vulnerabilities that exist in the wild. I became curious as to how well the scanners actually do audit these test applications, and how other vendors' scanners would work against them.

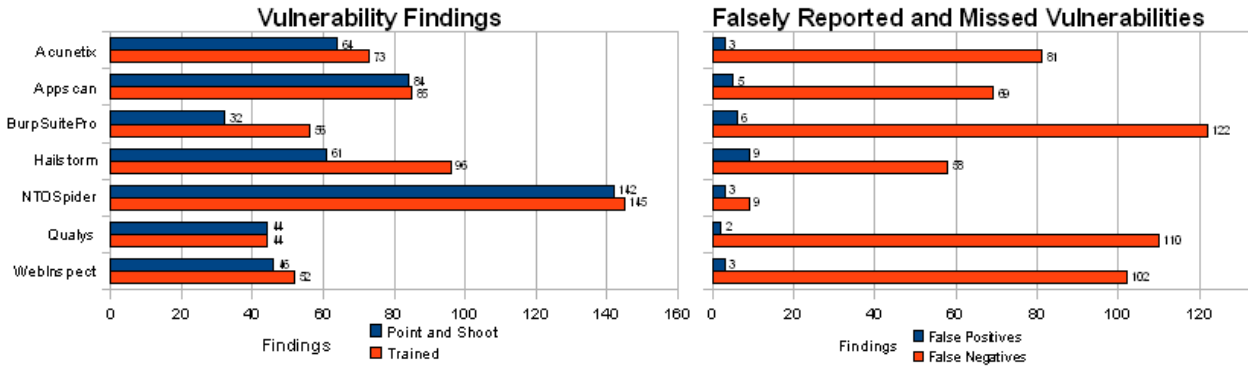
I decided to find out by running each vendor's scanner against each of the vendor's test sites and comparing the results. The assumption would be that each vendor would do the best against their own test site and the question would be which vendor would get 2nd place the most often. Part of the purpose of doing it this way is that it is reproducible by anyone with a copy of one of the scanners. The collected data is being made freely available for anyone to review and re-create.

Additionally the amount of time required to make good use of the scanners was of interest. So each scanner was run in 'Point and Shoot' and then again after being 'Trained' to know all the links and how to populate all the forms. These test sites are fairly small, most being in the 10-20 link range, with one or two in the 75-100 page range. For larger sites the training time could be extrapolated based on the observations in this study (assuming that the full inventory of pages and forms is known to the auditor).

Summary of Results

The full results of the testing are going to be analyzed in further detail later in the report. I will start off with some of the overview of the data and initial conclusions.

When looking at the results, there are a number of ways to look at the data. Rather than focus on code coverage as in the first report, this time the focus is on comparing the results of the scanners against each other at the heart of what these scanners are supposed to be doing: **finding vulnerabilities**. A review of the list of "possible" vulnerabilities that were found/missed offers up some rather interesting conclusions.



Based on all of the criticism of my first report, we should expect to see big differences between 'Point and Shoot' and 'Trained' scans, but it turns out that there are **only moderate improvements gained from normal training**. The one exception to this was with Cenzic Hailstorm which did improve dramatically when a solid effort was made to understand and apply all the configuration variables required for training. The findings from the first report which showed NT OBJECTives' NTOSpider with the most findings, followed by IBM Appscan and then HP WebInspect, remain consistent. In fact, WebInspect came in dead last even with the newcomers to this analysis on the software side (Acunetix, BurpSuitePro, Cenzic) and only managed to do a little better than the new Qualys offering.

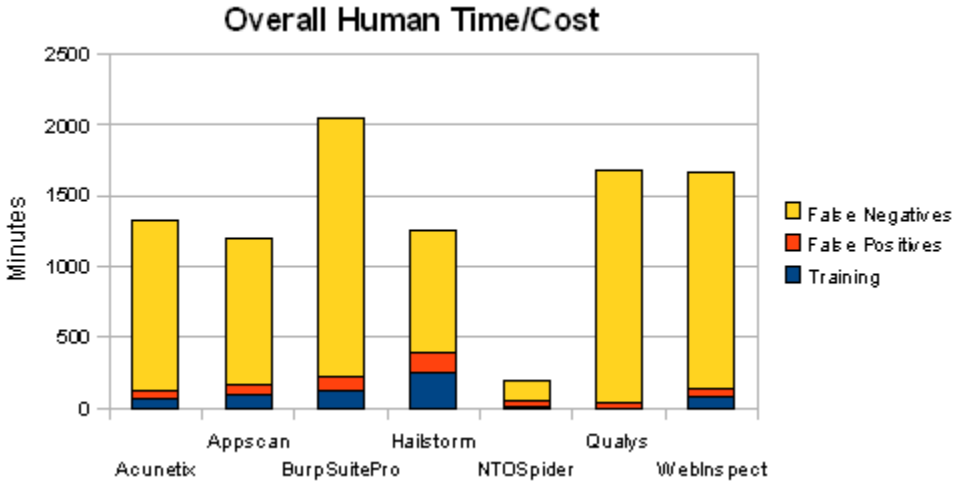
The False Positive rates were much less significant this time due to the methodology chosen which focused only on the big impact vulnerabilities (listed in Methodology section). In reality most of the scanners had many additional False Positives outside the categories included and were thus not counted.) NTOSpider remained among the best at avoiding False Positives along with WebInspect and Acunetix. **It is interesting to note that the scanners that miss the most vulnerabilities tended to also report the highest numbers of False Positives.** So not only do they miss more vulnerabilities, they waste more time when forced to weed out false positives.

When looking at the scan times, the tests show that the fastest scan times are in this order: BurpSuitePro, Qualys, WebInspect, NTOSpider, Appscan, Hailstorm, and then Acunetix as the slowest. However, to a large extent the amount of time the scanner takes to do the scan is less relevant because the most limited resource to a security professional is *human time*.

When looking at the amounts of time involved for the human involved to run a scan we generally think about the amount of time to configure and 'Train' the scanner, which varied a substantial amount among the scanners. The first scan to be run for each scanner was 'Point and Shoot', and based on the results and observed coverage the 'Training' was undertaken in accordance with what appeared to be needed.

Every effort was made during the training process to configure the scanners in every possible way to ensure that it could get its best results. This took time in reading the docs, and consulting with a number of experts with the tool and sometimes with the vendors themselves. The 'Training' time does not include this extra time to learn the tools, but is only for the amount of time actually interacting with the scanner, and should reflect the results you would get when a professional is proficient in the use of the specific scanner.

The final step was to take the amount of time needed to train the tool to get its best possible results, and then take into account the False Positive and False Negative numbers. A False Positive wastes time, during the vetting of the results. In addition, a general high rate of False Positives creates additional lack of trust in the results, which causes additional scrutiny of the result. False Negatives also costs time, due to the fact that a security professional must rely less on the tool and spend time doing more manual assessment of the application, which ultimately reduces the worth of the automated tool.



By applying a simple formula (described in Methodology) to calculate the cost of these False Positives and False Negatives to calculate the Overall Human Time/Cost we can take a more realistic look at the overall cost in human time that would be required to do an audit that would give us 99% confidence that due diligence was provided.

Methodology

In order to cover as many bases as possible it was decided to run each scanner in two ways:

1. **Point and Shoot (PaS):** This includes nothing more than run default scanning options and provide credentials if the scanner supported it and the site used any.
2. **Trained:** This includes any configurations, macros, scripts or other training determined to be required to get the best possible results. As needed help was requested from the vendors or from acquaintances with expertise in each scanner to make sure that each was given all possible opportunity to get its best possible results. This required days of Gotomeetings/Webexes with the "experts" helping tweak the scanners as well as they

could. The only scanner that was not trained was Qualys due to the fact that it is a managed service offering and training is not a normally available option.

In this review, the number of scanners involved was increased (alphabetical order)

- Acunetix Web Security Scanner (v6.5.20091130) from Acunetix
- Appscan (v7.8.0.2.891) from IBM
- BurpSuitePro (v1.3) from Portswigger.com
- Hailstorm (v6.0 build 4510) from Cenzic
- NTOSpider (v5.0.019) from NT OBJECTives
- Qualys Web Application Scanning from Qualys
- WebInspect (v8.0.753.0) from HP

(Note: WhiteHat Security declined to participate)

Each scanner is different in the types of server attacks it can perform, such as port scanning, application detection, and 'known vuln checking' as examples. For the purposes of this report, the types of vulnerabilities that were counted were those that are useful against custom applications, and which most users care about. These are:

- Authentication Bypass or Brute forcing
- SQL Injection / Blind SQL Injection
- Cross Site Scripting / Persistent Cross Site Scripting
- Command Injection
- XPath Injection
- SOAP/AJAX Attacks
- CSRF / HTTP Response Splitting
- Arbitrary File Upload attacks
- Remote File Include (PHP Code Injection)
- Application Errors (only those with debugging data useful for attacking)

The vulnerabilities were recorded to cross reference which scanners found which vulnerabilities in a simple format to track and compare the results.
I then created a complete listing of vulnerabilities discovered by each tool, and then manually verified their authenticity to compile a list of the overall "possible" vulnerabilities. This resulted in a fairly high degree of confidence that the extent of False Positives *and* False Negatives numbers was complete.

In order to determine the amount of human time that was needed to generate quality results the following formula was used:
Training time + (# False Positives * 15min) + (# False Negatives * 15min)
False Positive: Each finding takes time to confirm, and that time is a loss in the case of a False Positive.
False Negative: Higher rates of False Negatives reduces the auditors confidence in the tool, which demands additional manual assessment to be undertaken.
* The False Negative multiplier can easily be considered too low, but this number was use to keep it simple.

For the purpose of clarity, it should be pointed out that the Qualys testing was done in a different manner from the other tools. Having access to the other tools, it was possible to run them in trained and untrained modes. With Qualys, the sites to be scanned were ordered and the reports received. In theory, Qualys could have "gamed" the results (by hand testing the sites and inputting data, as opposed to using the tool). There is no reason to believe that was the case based on: 1) the reputation for honesty of my contact at Qualys and 2) their results which were tied for last (their web expert is certainly capable of doing much better should he have decided to game the results). Having said that, this paragraph needs to be included for full disclosure.

Detailed Results

After the extensive debates over the last report, it was clear that more detailed records of the findings would be included in this report. The full spreadsheet is included as Appendix 1. This section will cover in more detail the results and the experiences/opinions gained during the running and review of the scans.

When looking at the details its best to look at the grand total summary.

Overall Summary

			Acunetix		Appscan		BurpSuitePro		Hailstorm		NTOSpider		Qualys	WebInspect	
			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
Grand Totals		**Vuln's Found**	**64**	**73**	**84**	**85**	**32**	**56**	**61**	**96**	**142**	**145**	**44**	**46**	**52**
		% Found	**41.56%**	**47.40%**	**54.55%**	**55.19%**	**20.78%**	**36.36%**	**39.61%**	**62.34%**	**92.21%**	**94.16%**	**28.57%**	**29.87%**	**33.77%**
Valid Vulns	154	**Vuln's Missed**	**90**	**81**	**70**	**69**	**122**	**98**	**93**	**58**	**12**	**9**	**110**	**108**	**102**
Pages	315	**% Missed**	**58.44%**	**52.60%**	**45.45%**	**44.81%**	**79.22%**	**63.64%**	**60.39%**	**37.66%**	**7.79%**	**5.84%**	**71.43%**	**70.13%**	**66.23%**
		FP's Reported	3	3	5	3	2	6	9	7	3	3	2	3	2
		Scan Time	8:33	10:44	6:54	6:18	0:42	1:49	2:31	9:28	8:03	7:45	1:28	2:53	4:18
		Training Time	N/A	**1:10**	N/A	**1:30**	N/A	**2:05**	N/A	**4:10**	N/A	**0:05**	N/A	N/A	**1:25**
		Total Time	8:33	11:54	6:54	7:48	0:42	3:54	2:31	13:38	8:03	7:50	1:28	2:53	5:43

From this we get a good overview of what was generally experienced across all of the scanning that was performed during this analysis.

Entering this project it was assumed that each scanner would do the best against their own website and that the task would be in looking to find out who would consistently come in second best, and would therefore be the top scanner. These assumptions were not radically off, and most vendors did do very well against their own test sites, but they did not always win, and ended up simply missing legitimate vulnerabilities on their own test sites that other scanners could find.

Perhaps the most interesting result is the performance of WebInpect vs. their own website. These sites are made available to customers to show the power of the scanner during pre-sale evaluations. The companies are obviously aware of most, if not all, the vulnerabilities on the site because they were created by the vendor to be vulnerable. Presumably, the software companies

have an incentive to make sure that their scanner performs very well against their test site. They should also be a part of the vendor's QA process. WebInspect missed half of the vulnerabilities on its own test site even though every effort was made to train it to each of the pages I knew had vulnerabilities on them. Without belaboring a point made elsewhere, web application scanners are highly complex pieces of software that are easy to break in development if there is not a strong engineering team with good continuity. WebInspsect's false negatives against their own site are a significant cause for concern. Accunetix also missed 31% of the vulnerabilities against its two test sites.

Overall findings of the Scanners, in alphabetical order

Acunetix

Pros: Accunetix was a close third behind Appscan after being trained to find every link.
Cons: Accunetix missed 53% of the vulnerabilities even after being trained to know all of the pages. As mentioned previously, on their own test site, Accuntix missed 31% of the vulnerabilities after training and 37% without training. This is a significant cause for concern as they should be aware of the links vulnerabilities on their own site and be able to crawl and attack them. These test sites are relatively small; in any site that cannot be completely crawled manually, testers should be wary of relying exclusively on Accunetix given the weakness of its crawler.
Support: The staff at Acunetix is very responsive and was helpful with keeping their test sites up and resetting them as needed. When help was needed to understand how to best train the scanner using manual crawling, they promptly provided clear documentation on how to use the various included tools to accomplish the task.
Review: Accunetix lagged the industry leaders in point and shoot mode, giving rise to concerns about running it without significant training. If it is trained to find every link, it is a close third to Appscan.

Appscan

Pros: A high quality scanner with acceptable results on most sites. It performed well in 'point and shoot', better than all the scanners except NTOSpider.
Cons: JavaScript crawling was not as effective as would be desired (this determined by the URLs and vulnerabilities that it missed). During the testing Appscan had numerous scans crash or hang, which caused delays. Appscan missed 45% of vulnerabilities, even after the tool was trained to know all of the links.
Support: Appscan as a scanner I use regularly required little support for the study.
Review: Appscan is solid and seasoned scanning tool and while it did not top the study, it always delivers consistent and reliable results. In in Point and Shoot scanning, it came in as the clear second place solution.

BurpSuitePro

Pros: As a manual pen-testing tool, it is top rated. At its price point, its hard to argue against having it in your toolkit.
Cons: BurpSuitePro missed 64% of the vulnerabilities even after being trained to know all of the pages. It had the second biggest delta between trained and untrained results. BurpSuitePro completely lacks any JavaScript support which is a very big limitation. It also lacks any automated form population solution, and simply prompts the user to fill out any form it comes up to, which on many sites would be quite a significant effort. These test sites are all very small, which made manually crawling them fairly easy to do in a short time frame, but when scanning sites with thousands of pages, it would be entirely impracticable.
Support: No official support available
Review: BurpSuite is well recognized as a best of class hacking proxy. It is a useful companion to a full commercial tool. (Note: NTOSpider has recently added integration with BurpSuite to allow users to manually dig deeper into automated findings)

Hailstorm

Pros: The Cenzic web application security scanner has some very positive benefits to the web application security tester. It has great accuracy when trained effectively. Most of its attacks are highly customizable and configurable. It has a form training module that allows for fine grained control over the types of parameters that are submitted and can recollect these for later use. It has a modular architecture which allows different types of spidering and manual traversals to be combined with highly customized attacks. It was definitely the scanner with the most configuration options that could actually make a difference in the outcome of an assessment.
Cons: Hailstorm missed 38% of the vulnerabilities even after being trained to know all of the pages, but had missed 60% untrained. Cenzic scanner was the most challenging to configure for effective basic scans. It took 2-3 times longer to train for an effective scan as compared to most other scanners in its class. The Cenzic scanner is definitely geared for use by the more seasoned pen tester as shown from its numbers in the point and shoot category.
Support: The staff has been very helpful through the entire process, including answering calls.
Review: In general the observations of Hailstorm show that scanning even well understood and simple web applications requires a fairly knowledgeable understanding of the scanner. Human intervention is frequently required to get satisfactory results.

NTOSpider

Pros: NTOSpider was the most accurate scanner, finding over twice as many vulnerabilities as the average competitor even without training was able to discover 92% of the vulnerabilities, compared to the closest competition which was only able to find 55%. Once trained it increased to 94%, compared to the closest competition which was only able to find 62%. Great for fully automated scans, and now has better interface and manual training support.
Cons: Still needs work on the manual training features and possibly with scan times.

Support: The staff at NT OBJECTives was very helpful and responsive during the course of this study. Given their single focus, it is fairly easy to get support from the employees who work on the technology, as opposed to navigating a help desk.
Review: As clearly the leader in terms of quality results, NTOSpider performed very well. The results make a great case for using NTOSpider as the first choice for automated scanning.

Qualys

Note: For the purpose of clarity, it needs to be pointed out that the Qualys testing was done in a different manner than the other tools. See Methodology for details.
Pros: Because Qualys is a service, it is the ultimate point and shoot. You place your order and they deliver a report.
Cons: Although Qualys' results appear to be in line with the bottom tier of scanners, there is a significant concern. Qualys missed 39 out of 42 vulnerabilities against Webscantest (NTO's test site). Additional analysis reveals that many of these vulnerabilities require the tool to have a moderate level of JavaScript support. Qualys' materials note that their "Embedded web crawler parses HTML and **some** JavaScript to extract links" (emphasis mine). Based on these results, Qualys JavaScript parsing capabilities are extremely rudimentary. My contact made it clear that the JavaScript support was limited and that they would have an update out in a few months that includes better Javascript support. After some discussions with toolmakers, advanced JavaScript (much less AJAX) support is one of the most difficult things to add. Given the significant amount of JavaScript used by modern websites, the use of Qualys should be limited to websites with little or no JavaScript. It is also unclear what manual crawling solutions are available to help 'Train' the scanner for your site.
Support: Given that all that was done was to order the scans and download the results, there is no comment on Qualys' support.
Review: The Qualys web application scanner was competitive with the weaker tools in the market. Its crawling is still hampered by extremely limited JavaScript support; this is a significant problem given that it is intended to be a fully automated managed service.

WebInspect

Pros: The interface for reviewing the scan data is very well designed.
Cons: Poor vulnerability finding results, and had the worst score in this review. WebInspect missed 66% of the vulnerabilities, even after being trained to know all of the pages. They missed 42% of the vulnerabilities on their own test site after being trained and 55% before training. The manual training features are overly complicated and took a number of hours to learn how to do simple tasks. During the testing it had numerous scans crash or hang, which caused delays. All of these issues point to significant problems with maintaining quality post-Spi Dynamic's acquisition by HP.
Support: Difficult to reach anyone. Required help from colleagues and acquaintances to get questions answered.
Review: The apparent problems were very surprising for the industry market share leader. Many enterprises have been using WebInspect for years. These results bring into serious question its

abilities to find the latest vulnerabilities in modern websites; users of this tool should seriously consider re-evaluating their reliance on it as a method for securing their web applications.

Implications

The scanning vendors have spent a significant amount of time discovering a range of web application vulnerabilities both by independent research and by getting information from customers. As a whole, these vendor websites create a meaningful testbed to evaluate the performance of web application scanners. Some vendors will have the view that this is not an optimal way of looking at things, but this is a valid baseline with well understood vulnerabilities and the results can be validated fairly straightforwardly.

Some readers of this study may inquire why scans were not performed against some of the web applications created for teaching purposes (e.g. webgoat and hackme bank). First, these were not designed to mimic the functionality of real web applications but are intended for use in teaching a human how to perform an audit. The vendor test sites are more representative of the types of behaviors they would see in the wild. Second, some of the vendors are aware that users test against these sites and have pre-programmed their tools to report the vulnerabilities that they have already discovered. It is sort of like getting a copy of the test beforehand and memorizing that the answers are d,c,b,a, etc. as opposed to learning the material. The scanner may discover vulnerabilities on these sites but this has no predictive value for how it will perform for a user in testing their own sites.

I would also like to discuss this study in light of how it relates to a normal scanner evaluation. Web scanners will obviously have different results on different websites. For this reason, it is important to test the scanners against a range of websites with different technologies and vulnerabilities. Although NTOSpider was always at or near the top, results varied greatly by web application. In order to eliminate the effects of luck with small sample sizes, I decided to have at least 100 vulnerabilities in this test. Roughly 120 hours of work, plus access to all the scanners and experts in each to help, was put into this study, which may not be an option for many enterprises. Having said that, evaluating these tools on a small sample size of vulnerabilities can be a bit of a crap shoot. This is not to say that evaluators should not try the tools in their evaluations. But their results should be considered along with industry studies. One can get a sense of the feel of the tool in an evaluation - accuracy requires a larger investment of time. This is analogous to buying a car - you might get the feel of the vehicle from driving it but you should rely on Consumer Reports for certain things that may not be apparent during the test drive such as how well the engine performs over time (and certainly the crash test results).

Conclusion

The results of this study will be surprising to many. Even when web application scanners are directed to the vulnerable pages of a website, there is a significant discrepancy in the number of findings. Again, these results should not be surprising given the great difficultly of achieving accurate results over an infinite target space of custom web applications. This is a lot harder

problem than network scanning. These results should cause security professionals to have significant reason for concern if they are relying on one of the less accurate tools. There is a good chance that they are missing a significant number of vulnerabilities. The vulnerability results with the analysis of the time/cost involved in False Positive and False Negative findings should highlight additional areas of interest and consideration when picking a scanner. Given the large number of vulnerabilities missed by tools even when fully trained (56% when NTOSpider is eliminated from the results) it is clear that accuracy should still be the primary focus of security teams looking to acquire a tool.

The numerous crashes that I experienced with Appscan and WebInspect are also an issue that should be considered. As mentioned earlier, these are relatively small sites. The risk of a crash preventing completion of a scan will increase significantly with larger scans.

The data speaks for itself, and I was surprised that my previous report was largely validated by what I saw during this analysis and I was impressed by the results of NTOSpider with an excellent rate of vulnerability discovery, low false positives and terrific automation. For manual auditing, I was very impressed with BurpSuitePro which at roughly $200 is clearly a worthy tool to have in my toolkit. The biggest disappointment had to be with HP WebInspect which performed below my expectations. These results showed that it is not the size of marketing budgets that produce a better product.

Scanners with big deltas between trained and untrained results (Hailstorm, BurpSuitePro and Acunetix) can provide good results, but may require more effort to achieve them.

Response to my 2007 Study

In October 2007, I published a study, “Analyzing the Effectiveness and Coverage of Web Application Security Scanners"; in which I compared 3 commercial web application scanners, Appscan, NTOSpider and WebInpsect. The scanners were deployed in a 'Point and Shoot' method (i.e. I relied on their crawlers and did not point them to any areas of the websites being tested). I reported results for crawled links, application code functionality exercised (as monitored by Fortify Tracer) and vulnerability findings (both verified positive and false negatives). The results, as summarized in Appendix 2, showed that NTOSpider had far better coverage and vulnerability assessment than both Appscan and WebInspect.

I believe that the findings demonstrated that because of the nature of web applications, there can be a wide divergence in scanning results based on the quality of the scanner and/or specific functionality employed by the web application being scanned. Web application scanning is a much more difficult task than network scanning because most web applications are custom and scanners must crawl and attack them like a human, as opposed to searching for signatures, as network scanners do.

There was a significant amount of criticism of the results. After discussing the 2007 paper with numerous security professionals, I believe that the paper highlighted a significant fault line

within the security community. Broadly speaking, there are two groups in the web application testing community.
Group 1: Uses scanners in a more or less 'point and shoot' manner and relies on the scanners' crawler and automation to exercise the site's functionality within minimal or no human guidance. Their reasons for this include 1) they lack the time to spend training the scanner, 2) they want a repeatable result for audit purposes that is separate from the skill of a particular tester and 3) they believe that point and shoot results are sufficient to achieve the level of security testing on websites of the complexity that they are testing.

Group 2: Believes that scanning in a point and shoot manner is insufficient. They feel that given the complexity of modern websites, no automated tool can do an adequate job of testing a website without substantial human guidance. They often believe that scanners should be an adjunct to human testing and should be used to run a large number of easy attacks to get easy to find vulnerabilities ("low hanging fruit") and that human testers are required to get more difficult to find vulnerabilities. Members of Group 2 were the strongest critics of my original study.

Without opening up this can of worms again, I think that it is important to note that it is, in a sense, a pointless debate because regardless of the merits of either side, testers are going to fall into Group 1 or Group 2 or somewhere in the middle depending on their needs and skill sets. The point of this follow-up study is to address a criticism of Group 2. Group 2 argued that the 2007 study was not useful because I did not train the scanners (i.e. walk them through the websites that I scanned). If I had done this they claim that my results would have been different. This is certainly theoretically possible and was part of the impetus behind this second study.

Biography

Larry Suto is an independent consultant who has consulted for companies such as Wells Fargo, Pepsico, Kaiser Permanante, Charles Schwab and Cisco during his time with Strategic Data Command Inc. based in Oakland, CA. He specializes in enterprise security architecture, risk management, software quality analysis from a security perspective and RF security. Larry has been active in the industry for over twelve years and has worked with many Fortune 500 companies around the country. Recently his research has included introducing software testing and quality assurance techniques into the security engineering process in order to understand how the
effectiveness of security tools and processes can be measured with more accurate and quantifiable metrics.

Larry can be reached at larry.suto@gmail.com

Appendix 1:

Scan details and tracking data

Overall Summary

			Acunetix		Appscan		BurpSuitePro		Hailstorm		NTOSpider		Qualys	WebInspect	
			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
Grand Totals		**Vuln's Found**	64	73	84	85	32	56	61	96	142	145	44	46	52
		% Found	41.56%	47.40%	54.55%	55.19%	20.78%	36.36%	39.61%	62.34%	92.21%	94.16%	28.57%	29.87%	33.77%
Valid Vulns	154	**Vuln's Missed**	90	81	70	69	122	98	93	58	12	9	110	108	102
Pages	315	**% Missed**	58.44%	52.60%	45.45%	44.81%	79.22%	63.64%	60.39%	37.66%	7.79%	5.84%	71.43%	70.13%	66.23%
		FP's Reported	3	3	5	3	2	6	9	7	3	3	2	3	2
		Scan Time	8:33	10:44	6:54	6:18	0:42	1:49	2:31	9:28	8:03	7:45	1:28	2:53	4:18
		Training Time	N/A	1:10	N/A	1:30	N/A	2:05	N/A	4:10	N/A	0:05	N/A	N/A	1:25
		Total Time	8:33	11:54	6:54	7:48	0:42	3:54	2:31	13:38	8:03	7:50	1:28	2:53	5:43
Acunetix AspNet		**Vuln's Found**	6	6	6	6	2	5	5	5	7	7	5	4	4
		% Found	85.71%	85.71%	85.71%	85.71%	28.57%	71.43%	71.43%	71.43%	100.00%	100.00%	71.43%	57.14%	57.14%
Valid Vulns	7	**Vuln's Missed**	1	1	1	1	5	2	2	2	0	0	2	3	3
Pages	10	**% Missed**	14.29%	14.29%	14.29%	14.29%	71.43%	28.57%	28.57%	28.57%	0.00%	0.00%	28.57%	42.86%	42.86%
		FP's Reported	2	2	0	0	0	1	0	0	0	0	2	1	1
Acunetix TestPHP		**Vuln's Found**	16	18	11	10	9	16	10	10	27	27	10	9	10
		% Found	57.14%	64.29%	39.29%	35.71%	32.14%	57.14%	35.71%	35.71%	96.43%	96.43%	35.71%	32.14%	35.71%
Valid Vulns	28	**Vuln's Missed**	12	10	17	18	19	12	18	18	1	1	18	19	18
Pages	34	**% Missed**	42.86%	35.71%	60.71%	64.29%	67.86%	42.86%	64.29%	64.29%	3.57%	3.57%	64.29%	67.86%	64.29%
		FP's Reported	0	0	0	0	2	2	3	3	1	1	0	0	0
Cenzic Crackme		**Vuln's Found**	3	2	12	12	2	3	11	14	11	14	5	3	4
		% Found	17.65%	11.76%	70.59%	70.59%	11.76%	17.65%	64.71%	82.35%	64.71%	82.35%	29.41%	17.65%	23.53%
Valid Vulns	17	**Vuln's Missed**	14	15	5	5	15	14	6	3	6	3	12	14	13
Pages	28	**% Missed**	82.35%	88.24%	29.41%	29.41%	88.24%	82.35%	35.29%	17.65%	35.29%	17.65%	70.59%	82.35%	76.47%
		FP's Reported	0	0	0	0	0	1	3	2	0	0	0	0	0
HP zerowebappsecurity		**Vuln's Found**	9	12	10	9	11	11	9	25	31	31	11	14	18
		% Found	29.03%	38.71%	32.26%	29.03%	35.48%	35.48%	29.03%	80.65%	100.00%	100.00%	35.48%	45.16%	58.06%
Valid Vulns	31	**Vuln's Missed**	22	19	21	22	20	20	22	6	0	0	20	17	13
Pages	89	**% Missed**	70.97%	61.29%	67.74%	70.97%	64.52%	64.52%	70.97%	19.35%	0.00%	0.00%	64.52%	54.84%	41.94%
		FP's Reported	0	0	0	0	0	0	0	0	0	0	0	0	1
IBM Testfire		**Vuln's Found**	7	12	20	20	6	6	8	13	21	21	10	7	9
		% Found	29.17%	50.00%	83.33%	83.33%	25.00%	25.00%	33.33%	54.17%	87.50%	87.50%	41.67%	29.17%	37.50%
Valid Vulns	24	**Vuln's Missed**	17	12	4	4	18	18	16	11	3	3	14	17	15
Pages	32	**% Missed**	70.83%	50.00%	16.67%	16.67%	75.00%	75.00%	66.67%	45.83%	12.50%	12.50%	58.33%	70.83%	62.50%
		FP's Reported	0	0	0	0	0	0	0	0	0	0	0	0	0
NTO Webscantest		**Vuln's Found**	23	23	25	28	2	15	18	29	45	45	3	9	7
		% Found	48.94%	48.94%	53.19%	59.57%	4.26%	31.91%	38.30%	61.70%	95.74%	95.74%	6.38%	19.15%	14.89%
Valid Vulns	47	**Vuln's Missed**	24	24	22	19	45	32	29	18	2	2	44	38	40
Pages	122	**% Missed**	51.06%	51.06%	46.81%	40.43%	95.74%	68.09%	61.70%	38.30%	4.26%	4.26%	93.62%	80.85%	85.11%
		FP's Reported	1	1	5	3	0	2	3	2	2	2	0	2	0

Individual Test Site Results

Name:	Acunetix AspNet		Training Time	N/A	0:00	N/A	0:05	N/A	0:15	N/A	0:00	N/A	0:00	N/A	N/A	0:00
URL:	http://testaspnet.acunetix.com/		Scan Duration	0:49	0:49	0:14	0:14	0:15	0:43	0:45	0:45	0:54	0:54	0:16	0:50	0:50
User:	N/A		Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	N/A			**Acunetix**		**Appscan**		**BurpSuitePro**		**Hailstorm**		**NTOSpider**		**Qualys**	**WebInspect**	
Pages:	10			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**6**	**6**	**6**	**6**	**2**	**5**	**5**	**5**	**7**	**7**	**5**	**4**	**4**
	7		**Missed**	**1**	**1**	**1**	**1**	**5**	**2**	**2**	**2**	**0**	**0**	**2**	**3**	**3**
	6	**False Positives**	**Reported**	**2**	**2**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**2**	**1**	**1**
Status	**Vuln Type**	**File**	**Parameter**													
Verified	SQL	/Comments.aspx	id	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	SQL	/Comments.aspx	tbComment	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y		
Verified	SQL	/login.aspx	tbUsername	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y		
Verified	SQL	/ReadNews.aspx	id	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/Comments.aspx	tbComment	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS (persistent)	/Comments.aspx	tbComment			Y	Y					Y	Y			
Verified	XSS	/ReadNews.aspx	NewsAd	Y	Y							Y	Y		Y	Y
FP	SQL	/ReadNews.aspx	NewsAd						Y							
FP	SQL	/Comments.aspx	referer (Header)	Y	Y											
FP	SQL	/Comments.aspx	user-agent (Header)	Y	Y											
Duplicate	SQL	/Comments.aspx	accept-language (Header)	Y	Y											
Duplicate	SQL	/Comments.aspx	client-ip (Cookie)	Y	Y											
Duplicate	SQL	/Comments.aspx	x-forwarded-for (Header)	Y	Y				Y							
FP	Local File Inclusion	/Comments.aspx	tbComment												Y	Y
FP	XSS	/Comments.aspx	id											Y		
FP	XSS	/ReadNews.aspx												Y		

Name:	Acunetix TestPHP		Training Time	N/A	0:05	N/A	0:10	N/A	0:25	N/A	0:10	N/A	0:00	N/A	N/A	0:10
URL:	http://testphp.acunetix.com/		Duration	0:35	0:54	0:49	0:56	0:04	0:22	0:22	0:28	0:30	0:30	0:10	0:29	0:32
User:	maryblah		Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	mary123			**Acunetix**		**Appscan**		**BurpSuitePro**		**Hailstorm**		**NTOSpider**		**Qualys**	**WebInspect**	
Pages:	34			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**16**	**18**	**11**	**10**	**9**	**16**	**10**	**10**	**27**	**27**	**10**	**9**	**10**
	28		**Missed**	**12**	**10**	**17**	**18**	**19**	**12**	**18**	**18**	**1**	**1**	**18**	**19**	**18**
	6	**False Positives**	**Reported**	**0**	**0**	**0**	**0**	**2**	**2**	**3**	**3**	**1**	**1**	**0**	**0**	**0**
Status	**Vuln Type**	**File**	**Parameter**													
Verified	HTTP Res Splitting	/redir.php	r	Y	Y							Y	Y			
Verified	PHP Code Inject	/comment.php	phpaction	Y	Y											
Verified	XSS	/comment.php	name	Y	Y				Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/guestbook.php	name	Y	Y	Y	Y	Y	Y			Y	Y	Y		
Verified	XSS	/guestbook.php	text	Y	Y	Y	Y		Y			Y	Y	Y		
Verified	XSS	/guestbook.php	login (Cookie)		Y							Y	Y			
Verified	XSS	/listproducts.php	cat	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/listproducts.php	artist	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y		
Verified	XSS	/search.php	searchFor	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/secured/newuser.php	uuname	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y
Verified	XSS	/secured/newuser.php	uaddress									Y	Y			
Verified	XSS	/secured/newuser.php	uphone									Y	Y			
Verified	XSS	/secured/newuser.php	uemail									Y	Y			
Verified	XSS	/secured/newuser.php	ucc									Y	Y			
Verified	XSS	/secured/newuser.php	upass									Y	Y			
Verified	XSS	/secured/newuser.php	urname									Y	Y			
Verified	XSS	/userinfo.php	uaddress						Y			Y	Y			
Verified	XSS	/userinfo.php	urname						Y			Y	Y			
Verified	XSS	/userinfo.php	uemail				Y		Y	Y	Y	Y	Y			
Verified	XSS	/userinfo.php	ucc						Y			Y	Y			
Verified	SQL	/AJAX/infoartist.php	id	Y	Y							Y	Y		Y	Y
Verified	SQL	/AJAX/infocateg.php	id	Y	Y							Y	Y		Y	Y
Verified	SQL	/AJAX/infotitle.php	id	Y	Y							Y	Y		Y	Y
Verified	SQL	/artists.php	artist	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	SQL	/listproducts.php	cat	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	SQL	/listproducts.php	artist	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y		
Verified	SQL	/product.php	pic	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
Verified	SQL	/cart.php	addcart		Y	Y	Y		Y			Y	Y			
FP	XSS	/404.php	The URL Path							Y	Y					
FP	File Upload	/showimage.php	file							Y	Y					
FP	XSS	/showimage.php	file							Y	Y					
FP	OS Cmd Injection	/listproducts.php	cat					Y	Y							
FP	OS Cmd Injection	/listproducts.php	artist					Y	Y							
FP	SQL	/userinfo.php	uname									Y	Y			

Name:	Cenzic Crackme		Training Time	N/A	0:05	N/A	0:10	N/A	0:20	N/A	1:30	N/A	0:05	N/A	N/A	0:25
URL:	http://crackme.cenzic.com/		Scan Duration	2:01	2:47	0:28	0:33	0:09	0:23	0:21	4:31	3:33	3:15	0:17	0:07	0:37
User:	maryblah	admin	Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	mary123	adminm3		**Acunetix**		**Appscan**		**BurpSuitePro**		**Hailstorm**		**NTOSpider**		**Qualys**	**WebInspect**	
Pages:	28			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**3**	**2**	**12**	**12**	**2**	**3**	**11**	**14**	**11**	**14**	**5**	**3**	**4**
	17		**Missed**	**14**	**15**	**5**	**5**	**15**	**14**	**6**	**3**	**6**	**3**	**12**	**14**	**13**
	5	**False Positives**	**Reported**	**0**	**0**	**0**	**0**	**0**	**1**	**3**	**2**	**0**	**0**	**0**	**0**	**0**
Status	**Vuln Type**	**File**	**Parameter**													
Verified	SQL	/Kelew/php/accttransaction.php	FromDate			Y	Y			Y	Y	Y	Y			
Verified	SQL	/Kelew/php/accttransaction.php	ToDate			Y	Y			Y	Y	Y	Y			
Verified	SQL	/Kelew/view/updateloanrequest.php	txtAnnualIncome		Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtFirstName		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtLastName		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtSocialScurityNo		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtDOB		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtAddress		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtCity		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	drpState		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	txtEmail		Y							Y	Y			
Duplicate	SQL	/Kelew/view/updateloanrequest.php	drpLoanType		Y							Y	Y			
Verified	SQL	/Kelew/php/stock.php	symbol										Y			
Verified	SQL	/Kelew/php/stock.php	values										Y			
Verified	XSS	/Kelew/php/accttransaction.php	FromDate			Y	Y			Y	Y	Y	Y			
Verified	XSS	/Kelew/php/accttransaction.php	ToDate			Y	Y			Y	Y	Y	Y			
Verified	XSS	/Kelew/php/approveloanpagedetail.php	hRequestId							Y						
Verified	XSS	/Kelew/php/login.php	hLoginType	Y		Y	Y			Y	Y	Y	Y			Y
Verified	XSS	/Kelew/php/login.php	hUserType	Y		Y	Y			Y	Y	Y	Y	Y		
Verified	XSS	/Kelew/register/register.php	UserId	Y	Y	Y	Y			Y	Y		Y	Y		
Verified	XSS	/Kelew/view/updateloanrequest.php	txtAnnualIncome			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/Kelew/view/updateloanrequest.php	txtFirstName			Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/Kelew/view/updateloanrequest.php	drpLoanType								Y					
Verified	XSS	/Kelew/php/transfer.php	Amount			Y	Y				Y	Y	Y			
Verified	XSS	/Kelew/php/transfer.php	ToAccountNo			Y	Y				Y	Y	Y			
Verified	XSS	/Kelew/php/loanrequestdetail.php	hUserId								Y					
FP	SQL	/Kelew/view/rate.php	User-Agent (Header)								Y					
FP	SQL	/Kelew/php/loanrequestdetail.php	hRequestId							Y	Y					
FP	XSS	/Kelew/view/updateloanrequest.php	txtEmail							Y						
FP	XSS	/Kelew/view/updateloanrequest.php	txtTelephoneNo							Y						
FP	Command Injection	/Kelew/view/updateloanrequest.php	txtAnnualIncome						Y							

Name:	HP zerowebappsecurity		Training Time	N/A	0:10	N/A	0:20	N/A	0:15	N/A	1:45	N/A	0:00	N/A	N/A	0:15
URL:	http://zero.webappsecurity.com		Scan Duration	1:34	2:12	0:15	0:10	0:06	0:11	0:14	2:47	1:32	1:32	0:06	0:18	0:49
User:	admin		Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	admin			**Acunetix**		**Appscan**		**BurpSuitePro**		**Hailstorm**		**NTOSpider**		**Qualys**	**WebInspect**	
Pages:	89			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**9**	**12**	**10**	**9**	**11**	**11**	**9**	**25**	**31**	**31**	**11**	**14**	**18**
	31		**Missed**	**22**	**19**	**21**	**22**	**20**	**20**	**22**	**6**	**0**	**0**	**20**	**17**	**13**
	1	**False Positives**	**Reported**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
Status	**Vuln Type**	**File**	**Parameter**													
Verified	Authentication	/banklogin.asp (admin/admin)	login									Y	Y	Y		
Verified	SQL	/forgot1.asp	get								Y	Y	Y		Y	Y
Verified	SQL	/login1.asp	login	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/acctxferconfirm.asp	amount		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
Verified	XSS	/acctxferconfirm.asp	fromAcct		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
Verified	XSS	/acctxferconfirm.asp	toAcct		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
Verified	XSS	/banklogin.asp	err	Y	Y	Y		Y	Y		Y	Y	Y	Y		
Verified	XSS	/cookietest/ShowCookies.asp	FirstCookie (Cookie)					Y	Y		Y	Y	Y		Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	Keyed (Cookie)					Y	Y		Y	Y	Y		Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	Second (Cookie)					Y	Y		Y	Y	Y		Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	sessionid (Cookie)					Y	Y		Y	Y	Y		Y	
Duplicate	XSS	/cookietest/ShowCookies.asp	State (Cookie)					Y	Y		Y	Y	Y		Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	status (Cookie)					Y	Y		Y				Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	userid (Cookie)					Y	Y		Y	Y	Y		Y	Y
Duplicate	XSS	/cookietest/ShowCookies.asp	username (Cookie)					Y	Y		Y	Y	Y		Y	Y
Verified	XSS	/forgot2.asp	msg								Y	Y	Y		Y	Y
Verified	XSS	/join1.asp	address2								Y	Y	Y			
Verified	XSS	/join1.asp	country								Y	Y	Y			
Verified	XSS	/join1.asp	email								Y	Y	Y		Y	Y
Verified	XSS	/join1.asp	homephone								Y	Y	Y			
Verified	XSS	/join1.asp	house								Y	Y	Y			
Verified	XSS	/join1.asp	mobilephone								Y	Y	Y			
Verified	XSS	/join1.asp	msg								Y	Y	Y			
Verified	XSS	/join1.asp	name								Y	Y	Y			Y
Verified	XSS	/join1.asp	postcode								Y	Y	Y			
Verified	XSS	/join1.asp	street								Y	Y	Y			Y
Verified	XSS	/join1.asp	surname								Y	Y	Y			Y
Verified	XSS	/join1.asp	town								Y	Y	Y			Y
Verified	XSS	/login/login.asp	UserName	Y	Y							Y	Y		Y	Y
Verified	XSS	/login1.asp	login	Y	Y	Y	Y			Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/pcomboindex.asp	cboPage	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/pcomboindex.asp	referer (Header)	Y	Y			Y	Y			Y	Y			
Duplicate	XSS	/pcomboindex.asp	sessionid (Cookie)	Y	Y			Y	Y			Y	Y			
Duplicate	XSS	/pcomboindex.asp	State (Cookie)	Y	Y			Y	Y			Y	Y			
Duplicate	XSS	/pcomboindex.asp	user-agent (Header)	Y	Y			Y	Y					Y		
Duplicate	XSS	/pcomboindex.asp	userid (Cookie)	Y	Y			Y	Y			Y	Y			
Duplicate	XSS	/pcomboindex.asp	username (Cookie)	Y	Y			Y	Y			Y	Y			
Verified	XSS	/pformresults.asp	dbConnectString									Y	Y		Y	Y
Verified	XSS	/pformresults.asp	txtFirstName									Y	Y		Y	Y
Verified	XSS	/pformresults.asp	txtLastName									Y	Y		Y	Y
Verified	XSS	/plink.asp	a	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/plink.asp	c	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/rootlogin.asp	txtName	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
FP	SQL	/cfmerror.html	(none)													Y

Name:	IBM Testfire		Training Time	N/A	0:15	N/A	0:00	N/A	0:20	N/A	0:15	N/A	0:00	N/A	N/A	0:05
URL:	http://demo.testfire.net		Scan Duration	0:15	0:43	0:54	0:54	0:05	0:05	0:29	0:21	0:43	0:43	0:34	0:25	0:34
User:	jsmith		Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	Demo1234			**Acunetix**		**Appscan**		**BurpSuitePro**		**Hailstorm**		**NTOSpider**		**Qualys**	**WebInspect**	
Pages:	32			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**7**	**12**	**20**	**20**	**6**	**6**	**8**	**13**	**21**	**21**	**10**	**7**	**9**
	24		**Missed**	**17**	**12**	**4**	**4**	**18**	**18**	**16**	**11**	**3**	**3**	**14**	**17**	**15**
	0	**False Positives**	**Reported**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Status	**Vuln Type**	**File**	**Parameter**													
Maybe	Error/XPath Injection	/bank/queryxpath.aspx	SOAP: TextBox1		Y	Y	Y			Y						
Verified	SQL Auth Bypass	/bank/login.aspx	uid			Y	Y					Y	Y	Y		
Verified	SQL	/ (many pages)	amUserId (Cookie)			Y	Y	Y	Y			Y	Y			Y
Verified	SQL	/bank/account.aspx	listAccounts		Y	Y	Y			Y	Y	Y	Y			
Verified	SQL	/bank/login.aspx	passw	Y	Y	Y	Y	Y	Y		Y	Y	Y		Y	Y
Verified	SQL	/bank/login.aspx	uid	Y	Y	Y	Y	Y	Y		Y	Y	Y		Y	Y
Verified	SQL	/bank/login.aspx	btnSubmit			Y	Y									
Verified	SQL	/bank/transaction.aspx	after		Y	Y	Y					Y	Y	Y	Y	Y
Verified	SQL	/bank/transaction.aspx	before		Y	Y	Y					Y	Y	Y	Y	Y
Verified	SQL	/bank/transfer.aspx	debitAccount		Y	Y	Y					Y	Y	Y		
Verified	SQL	/bank/transfer.aspx	creditAccount		Y	Y	Y					Y	Y	Y		
Verified	SQL	/bank/ws.aspx	SOAP: creditAccount			Y	Y					Y	Y		Y	Y
Verified	SQL	/bank/ws.aspx	SOAP: debitAccount			Y	Y					Y	Y			
Verified	SQL	/bank/ws.aspx	SOAP: transferDate								Y					
Verified	SQL	/bank/ws.aspx	SOAP: transferAmount							Y	Y					
Verified	SQL	/subscribe.aspx	txtEmail	Y	Y	Y	Y				Y	Y	Y			
Verified	XSS	/bank/customize.aspx	lang		Y	Y	Y			Y	Y	Y	Y	Y		
Verified	XSS	/bank/login.aspx	uid	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Verified	XSS	/bank/transfer.aspx	creditAccount			Y	Y					Y	Y	Y		
Verified	XSS	/bank/transfer.aspx	debitAccount			Y	Y					Y	Y	Y		
Verified	XSS	/comment.aspx	name	Y		Y	Y	Y	Y	Y	Y	Y	Y		Y	Y
Verified	XSS	/notfound.aspx	aspxerrorpath							Y	Y	Y	Y			
Verified	XSS	/search.aspx	txtSearch	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		Y
Verified	XSS	/subscribe.aspx	txtEmail	Y	Y	Y	Y				Y	Y	Y			
Verified	XSS (DOM)	/disclaimer.htm	url (or DummyParam)							Y	Y	Y	Y			

Name:	NTO Webscantest		Training Time	N/A	0:35	N/A	0:45	N/A	0:30	N/A	0:30	N/A	0:00	N/A	N/A	0:30
URL:	http://www.webscantest.com/		Scan Duration	3:19	3:19	4:14	3:31	0:03	0:05	0:20	0:36	0:51	0:51	0:05	0:44	0:56
User:	testuser		Version	6.5.20091130		7.8.0.2.891		1.3		6.0 build 4510		5.0.019		PaS	8.0.753.0	
Pass:	testpass			Acunetix		Appscan		BurpSuitePro		Hailstorm		NTOSpider		Qualys	WebInspect	
Pages:	122			PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained
	Summary															
	Possible	**Valid Vulnerabilities**	**Found**	**23**	**23**	**25**	**28**	**2**	**15**	**18**	**29**	**45**	**45**	**3**	**9**	**7**
	47		**Missed**	**24**	**24**	**22**	**19**	**45**	**32**	**29**	**18**	**2**	**2**	**44**	**38**	**40**
	21	**False Positives**	**Reported**	**1**	**1**	**5**	**3**	**0**	**2**	**3**	**2**	**2**	**2**	**0**	**2**	**0**
Status	**Vuln Type**	**File**	**Parameter**													
Verified	App Errors / PA	/payment_analysis/checkdata_get.php	alpha_only	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	App Errors / PA	/payment_analysis/checkdata_get.php	letters_only	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	App Errors / PA	/payment_analysis/checkdata_get.php	number	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	App Errors / PA	/payment_analysis/checkdata.php	alpha_only	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	App Errors / PA	/payment_analysis/checkdata.php	letters_only	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	App Errors / PA	/payment_analysis/checkdata.php	number	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	Auth Testing	/login.php (admin/admin)	login					Y				Y	Y	Y		
Verified	Command Injection	/osrun/whois.php	domain	Y	Y	Y	Y		Y			Y	Y			
Verified	HTTP Res Splitting	/hrs/redir.php	q	Y	Y							Y	Y			
Verified	Remote File Include	/	incdir									Y	Y			
Verified	Arbitrary File Upload	/picshare/uploadpic.php	userfile									Y	Y			
Verified	Arbitrary File Upload	/picshare/upload.pl	file									Y	Y			
Verified	SQL	/login.php	login	Y	Y				Y		Y	Y	Y	Y	Y	
Verified	SQL	/datastore/getimage_by_id.php	id	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	SQL	/datastore/getimage_by_name.php	name	Y	Y	Y	Y				Y	Y	Y			
Verified	SQL	/datastore/search_by_id.php	id	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y
Verified	SQL	/datastore/search_by_name.php	name	Y	Y	Y	Y		Y	Y	Y	Y	Y			
Verified	SQL	/datastore/search_double_by_name.php	name	Y	Y				Y	Y	Y	Y	Y			
Verified	SQL	/datastore/search_get_by_id.php	id	Y	Y	Y	Y		Y	Y	Y	Y	Y		Y	Y
Verified	SQL	/datastore/search_get_by_name.php	name	Y	Y	Y	Y		Y	Y	Y	Y	Y			
Verified	SQL	/shutterdb/item.php	id									Y	Y			
Verified	SQL	/shutter_db/filter_by_name.php	filter						Y		Y	Y	Y			
Verified	SQL	/shutterdb/search_by_id.php	id				Y		Y		Y	Y	Y			
Verified	SQL	/shutterdb/search_by_name.php	name				Y		Y		Y	Y	Y			
Verified	SQL	/shutterdb/search_get_by_id.php	id				Y		Y		Y	Y	Y			
Verified	SQL	/shutterdb/search_get_by_id2.php	id				Y		Y		Y	Y	Y			
Verified	SQL	/shutterdb/search_get_by_id3.php	id				Y		Y		Y	Y	Y			
Verified	XSS	/ 404 handler (any random page)	referer (Header)			Y	Y	Y				Y	Y			
Verified	XSS	/bjax/servertime.php	msg	Y	Y							Y	Y			
Verified	XSS	/crosstraining/aboutyou2.php	fname	Y	Y	Y	Y		Y	Y	Y	Y	Y			
Verified	XSS	/crosstraining/aboutyou2.php	nick	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	XSS	/crosstraining/aboutyou2.php	returnto	Y	Y	Y	Y			Y	Y	Y	Y			
Verified	XSS	/crosstraining/reservation_submit.php	arrive_date	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y
Verified	XSS	/crosstraining/review.php	description			Y						Y	Y		Y	Y
Verified	XSS	/crosstraining/review.php	email			Y						Y	Y		Y	Y
Verified	XSS	/crosstraining/sitereviews.php	description	Y	Y	Y	Y			Y	Y	Y	Y		Y	Y
Verified	XSS	/crosstraining/sitereviews.php	email	Y	Y	Y	Y		Y			Y	Y		Y	Y
Verified	XSS	/login.php	login_error									Y	Y			
Verified	XSS	/myfiles/	xyz									Y	Y			
Verified	XSS	/soap/client1.php	proxyhost												Y	
Verified	XSS (persistent)	/crosstraining/reservation_history.php	arrive_date								Y	Y	Y			
Verified	XSS (persistent)	/crosstraining/reservation_history.php	fname			Y	Y					Y	Y			
Verified	XSS (persistent)	/crosstraining/reservation_submit.php	fname								Y	Y	Y			
Verified	XSS (persistent)	/crosstraining/review.php	description									Y	Y			
Verified	XSS (persistent)	/crosstraining/sitereviews.php	description			Y	Y			Y	Y	Y	Y			
Verified	XSS (persistent)	/crosstraining/sitereviews.php	email			Y	Y					Y	Y			
Verified	XSS	/picshare/upload.pl	kie)								Y			Y		
FP	Command Injection	/datastore/search_by_id.php	id						Y							
FP	Command Injection	/datastore/search_get_by_id.php	id						Y							
FP	HTTP Res Splitting	/hrs/redir_nv.php	q	Y	Y											
FP	local file inclusion	/crosstraining/sitereviews.php	submit												Y	
FP	SQL	/crosstraining/sitereviews.php	name													
FP	SQL	/crosstraining/sitereviews.php	description													
FP	SQL	/crosstraining/sitereviews.php	rating													
FP	SQL	/crosstraining/sitereviews.php	title													
FP	SQL	/crosstraining/review.php	email			Y										
FP	SQL	/crosstraining/review.php	description			Y										
FP	SQL	/crosstraining/review.php	name			Y										
FP	SQL	/payment_analysis/checkdata_get.php	number									Y	Y			
FP	SQL	/payment_analysis/checkdata.php	alpha_only							Y						
FP	SQL	/payment_analysis/checkdata.php	letters_only							Y						
FP	SQL	/payment_analysis/checkdata.php	number							Y		Y	Y			
FP	SQL	/osrun/whois_nv.php	domain								Y					
FP	SQL	/osrun/whois.php	domain								Y					
FP	XSS	/crosstraining/linkout.php	name			Y	Y									
FP	XSS	/crosstraining/reservation_submit.php	cn												Y	
FP	XSS (persistent)	/crosstraining/sitereviews.php	submit			Y	Y									
FP	XSS (persistent)	/crosstraining/sitereviews.php	name				Y									

Appendix 2:

Summary of Results from Analyzing the Effectiveness and Coverage of Web Application Security Scanners October, 2007

Closed Source - Internal Corporate Application

Scanner	Links Crawled	Database API	Web API	Total API	Vuln Findings	False Positives
NTOSpider	91	20	35	55	0	0
AppScan	113	17	24	41	0	0
WebInspect	91	17	23	40	0	0

Roller - Open Source Blogging platform

Scanner	Links Crawled	Database API	Web API	Total API	Vuln Findings	False Positives
NTOSpider	736	2	121	123	2	0
AppScan	129	1	98	99	0	5
WebInspect	663	1	68	69	1	10

OpenCMS - Open Source Customer Management application

Scanner	Links Crawled	Database API	Web API	Total API	Vuln Findings	False Positives
NTOSpider	3,380	47	158	205	225	0
AppScan	742	36	132	168	27	0
WebInspect	1,687	45	140	185	11	3

Totals

Scanner	Links Crawled	Database API	Web API	Total API	Vuln Findings	False Positives
NTOSpider	4,207	69	314	383	227	0
AppScan	984	54	254	308	27	5
WebInspect	2,441	63	231	294	12	13