



NT OBJECTIVES, INC.

# Innovating Application Security

[www.ntobjectives.com](http://www.ntobjectives.com)



## NT OBJECTives, Inc.

---

- Team of professionals focused on Web Application Security
  - Engineering team created FoundScan and NT version of TripWire
- Complete solution
  - Managed Scans
  - Software
  - Training
  - Remediation and Process Consulting



# Customers

---

- **Enterprises**

- Cisco
- Booz Allen Hamilton
- Department of Energy
- Deutsche Telekom
- Marketlive

- **Consulting**

- Accuvant
- Brintech
- Netraguard
- SECNAP
- Sentor
- Sogeti
- Telos

- **OEM**

- Veracode
  - eEye Digital Security
-



# Web Application VA vs. Network VA

	<b>Network VA</b>	<b>Web Application VA</b>
<b>Assessment Space</b>	Known IP port range	Near Infinite Must be discovered; from home page
<b>Unique targets</b>	Finite number of OS's, services and devices	Infinite unique Applications
<b>Vulnerability Responses</b>	Consistent to each device and vuln	Different for each Application
<b>Technological Complexity</b>	Signature-based	Heuristic intelligence



# Why Web Application Security Matters

---

- Gartner estimates over 70% of all website attacks target the application
  - “That’s Where The Money Is” Willie Sutton
  - Much lower investment in security architecture per program than operating systems and web server software
- Payment Credit Industry - Data Security Standards (PCI-DSS) require audit of applications
  - Fines for lack of compliance
- Gramm-Leach-Bliley - mandates safeguards
- State laws - require notification of data loss in California
  - Failure can result in criminal prosecution



## Choice Of Approaches

---

- 100% Automation as the goal – NT OBJECTIVES
  - No interactive/train mode until recently
  - Rapid response to bugs to improve accuracy and automation
- Training Required – Competitors
  - Creating Login Macros expected for each application
  - Train tool to avoid false positives
  - *"In order to achieve good results, web application scanners should be used in conjunction with manual security assessment, which requires close acquaintance with the web application and its different functionalities."* Ory Segal, Watchfire Director of Research



# Focus: Automation In Crawling

---

- Crucial to crawl the entire site
  - ***You cannot attack what you cannot crawl***
  - Basic Crawling Technology
    - Go to a start page
    - Get the HTML and search for links
    - Crawl the found links
    - Get more links
    - Keep going until you run out of links to crawl
  - Sounds simple... Not so fast
-



## Focus: Automation Complexities

- Brochureware Websites are easy to crawl
- Most webapps have challenging features
  - Generally designed for a human
  - Forms that validate data before going to the next step
  - JavaScript - Client side scripting
- Browsers have millions of hours of development time over many years to be functional. Scanners must replicate and automate much of this
- Most competing products cannot completely crawl complex sites

“The only application scanner that actually works on my complex site.”

**McGraw-Hill**



# Focus: Automation In Crawling JavaScript

- Not all links are easily parsed from within JavaScript functions. Look at this example:

```
<script>
  function gotoRequestQuote() { window.location.href='/getquote.asp'; }
  function gotoDept(input) { window.location.href='/'+input+'.asp'; }
</script>
<button type="button" onClick="javascript: gotoRequestQuote();">Request Quote</button>
<select onChange="javascript: gotoDept(this.value);">
  <option value="sales">Sales</option>
  <option value="support">Support</option>
</select>
```

- Links like **"/getquote.asp"** are easily found from parsing
- Links constructed by JavaScript can only be found when using advanced JavaScript analysis and execution as is the case for **"/sales.asp"** and **"/support.asp"**
- Manual crawling of JavaScript can take dozens of man hours



# Focus: Automation In Form Population

- HTML Forms need to be filled out properly

First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Address:	<input type="text"/>
State:	<input type="text"/>
Zip:	<input type="text"/>

- But, The input tags can be named anything

```
<table>  
  <tr><td>First Name: </td><td><input name="var1" /></td></tr>  
  <tr><td>Last Name: </td><td><input name="var2" /></td></tr>  
  <tr><td>Address: </td><td><input name="var3" /></td></tr>  
  <tr><td>State: </td><td><input name="var4" size="2" /></td></tr>  
  <tr><td>Zip: </td><td><input name="var5" size="5" /></td></tr>  
</table>
```

- Advanced Analysis of the Presentation Layer must be used to determine the correct data to enter into each input field.



## Focus: Automation In Logins

---

- Login Form identification
  - Humans are smart – we can identify a login form visually
  - Crawlers must be smart enough to identify a login form
    - **Many different login form layouts and field names**
    - **Sloppy HTML adds to the difficulty**
- Login process identification
  - Identify unsuccessful login vs successful login to confirm you have logged in
- Login complexities
  - Redirects, Cross-port redirects and Cross-domain redirects
  - Redirects to links that cannot be attacked
  - JavaScript used in login process
  - Intelligence to avoid logging out



## Focus: Session Management

---

- Servers change cookies periodically
- Browsers keep track of one cookie value at a time
  - Make a request, wait for it to come back, make another request
- Crawlers are multi-threaded
  - Send out multiple requests at once
  - Can have dozens of simultaneous outstanding requests



## Crawling: Benchmark Testing

- Larry Suto 2007 Study ([ha.ckers.org/](http://ha.ckers.org/))
  - First published statistical study of web application scanners
  - Independent Consultant (Cisco, Pepsi, Wells Fargo, Charles Schwab)
  - Hooked applications up to Tracer to test codebase coverage

	<b>NTOSpider</b>	<b>WebInspect</b>	<b>Appscan</b>
Links Crawled	4,207	2,441	984
Codebase Coverage	383	294 (23% less)	308 (20% less)



# Accuracy: False Positives/False Negatives

---

- Vulnerability responses are not consistent
  - Custom error pages vary greatly and can lead to false positives
- Requires advanced heuristics
  - Suppress **false positives**
  - Not create **false negatives**
  - Difficult balance, made easier with better logic
- False Positive Costs
  - Time to investigate
  - Reputational impact to security team/MSSP
- False Negative Costs
  - Still exposed
  - Fewer findings reduces perception of value



## Accuracy: Confirmation is the key

---

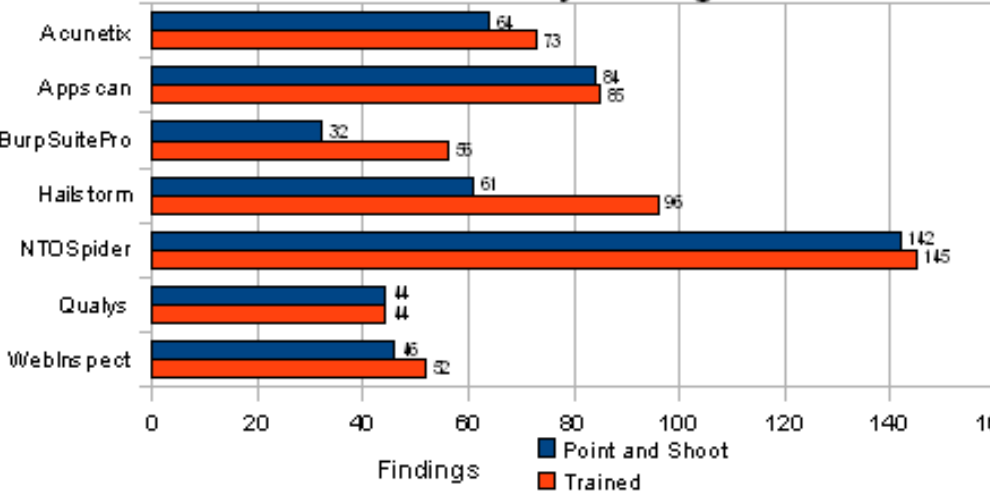
- Other scanners are too quick to claim they found a vulnerability
  - Request #1: product.asp?id=5
  - Request #2: product.asp?id=6
    - Is #2 different from #1? If so continue.
  - Request #3: product.asp?id=mod(11,6)
    - Does #3 match #1 response?
    - If so other tools claim they found a *Vulnerability*
    - But what if #1 and #3 had both simply resulted in a "Product Not Found" page? NTOSpider goes further...
  - Request #4: product.asp?id=mod(13,7)
    - Does #4 match #2 response? If so: **Confirmed Vulnerability**



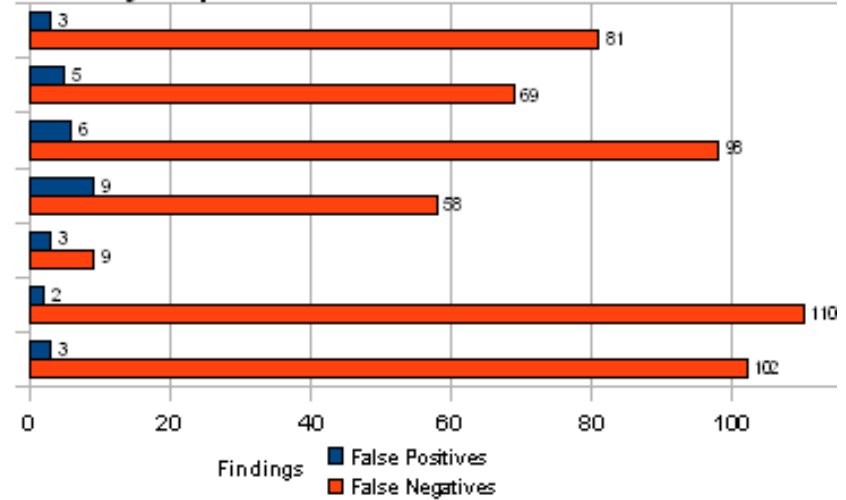
# Attacking: Benchmark Testing

- Larry Suto 2010 Study ([ha.ckers.org/](http://ha.ckers.org/))
  - Most recent published study of web application scanners

### Vulnerability Findings



### Falsely Reported and Missed Vulnerabilities





# Attacking: Benchmark Testing

- Larry Suto 2010 Study (ha.ckers.org/)

Overall Summary		Acunetix		Appscan		BurpSuitePro		Hailstorm		NTOSpider		Qualys	WebInspect		
		PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	Trained	PaS	PaS	Trained	
<b>Grand Totals</b>	<b>Vuln's Found</b>	64	73	84	85	32	56	61	96	142	145	44	46	52	
	<b>% Found</b>	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	<b>Vuln's Missed</b>	90	81	70	69	122	98	93	58	12	9	110	108	102
Pages	315	<b>% Missed</b>	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%
	<b>FP's Reported</b>	3	3	5	3	2	6	9	7	3	3	2	3	2	
	<b>Scan Time</b>	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	
	<b>Training Time</b>	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	
	<b>Total Time</b>	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	



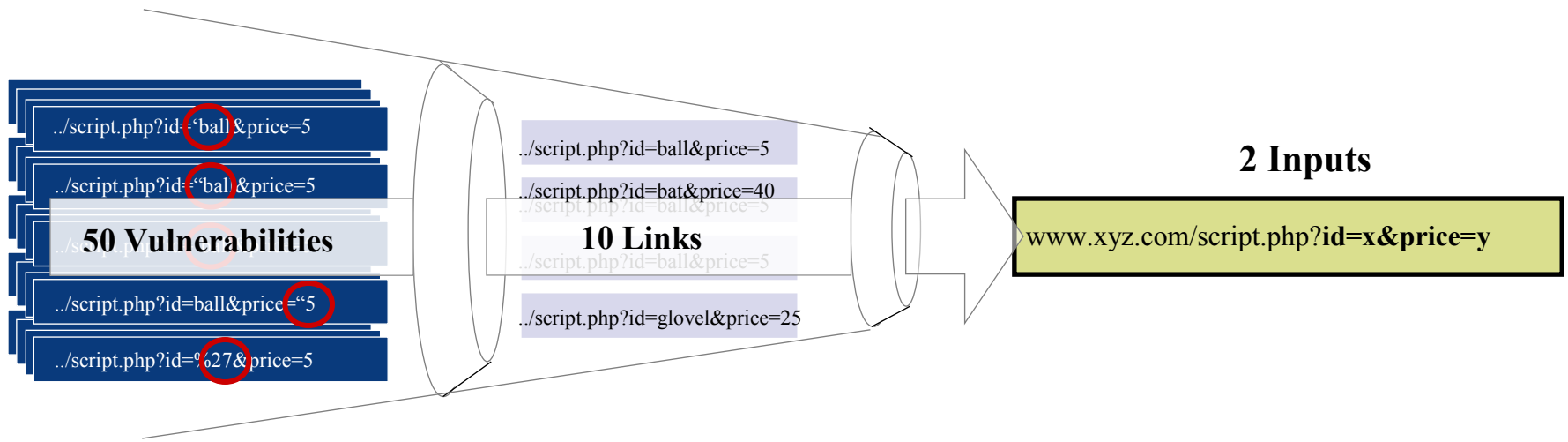
# NTOSpider Reporting

---

- Vulnerability identification is only first step
- Vulnerabilities must be
  - Verified
  - Communicated
  - Remediated
  - Re-Tested
- NTO Reports focused on facilitating assignment and remediation of vulnerabilities



# Consolidation of Root Causes



- NTOSpider consolidates numerous vulnerabilities into a handful of vulnerable inputs
- Easy to communicate source of problem to developers
- Facilitates prioritization, coordination and tracking of remediation efforts



# Reporting: Consolidation and Validation

**Cross Site Scripting** HIGH

Site: <http://scanme.ntobjectives.com:80>

[http://scanme.ntobjectives.com:80/vuln\\_site/crosstraining/review.php](http://scanme.ntobjectives.com:80/vuln_site/crosstraining/review.php) Root Causes #19 - 20: 2 parameters / 18 vulns HIGH

[http://scanme.ntobjectives.com:80/vuln\\_site/crosstraining/aboutyou2.php](http://scanme.ntobjectives.com:80/vuln_site/crosstraining/aboutyou2.php) Root Causes #21 - 22: 2 parameters / 12 vulns MED

URL: [http://scanme.ntobjectives.com:80/vuln\\_site/crosstraining/aboutyou2.php](http://scanme.ntobjectives.com:80/vuln_site/crosstraining/aboutyou2.php) Root Cause #21: fname MED

Vulnerable Parameter	Original Value	Method	Error	Validate
fname	John	POST		
<b>Attack Type</b>	<b>Attack Value</b>			
Unfiltered <script>	<script>eval(alert(String.fromCharCode(122,88,116,80,117,77,119,81)))</script>		<b>Successful XSS Attack</b> zdpurnwq OK	Validate
Unfiltered <script>	<<script>alert(String.fromCharCode(116,82,120,87,122,82,116,90));</script>		<b>Successful XSS Attack</b> trwzrtz OK	Validate
Unfiltered <iframe> src	<iframe src=zVpQzSwU></iframe>		Welcome <iframe src=zVpQzSwU></iframe> "" John. 	Validate
Unfiltered <script> src	<script src=yUmXrXzX <		Welcome <script src=yUmXrXzX < "" John. 	Validate
Unfiltered <script> src	<script src=sQkKWtU>		Welcome <script src=sQkKWtU> "" John. 	Validate
Unfiltered <iframe> src	<iframe src=xNkRyQKT <		Welcome <iframe src=xNkRyQKT < "" John. 	Validate

URL: [http://scanme.ntobjectives.com:80/vuln\\_site/crosstraining/aboutyou2.php](http://scanme.ntobjectives.com:80/vuln_site/crosstraining/aboutyou2.php) Root Cause #22: nick MED

Vulnerable Parameter	Original Value	Method	Error	Validate
nick	John	POST		
<b>Attack Type</b>	<b>Attack Value</b>			
Unfiltered <script>	<script>eval(alert(String.fromCharCode(118,76,108,84,115,87,111,81)))</script>		<b>Successful XSS Attack</b> vltswq OK	Validate
Unfiltered <script>	<<script>alert(String.fromCharCode(121,79,116,84,120,87,110,84));</script>		<b>Successful XSS Attack</b> yotbwnt OK	Validate
Unfiltered <script> src	<script src=xVyNyTnX>		Welcome John "<script src=xVyNyTnX>" John. 	Validate
Unfiltered <iframe> src	<iframe src=kVIMKTqO <		Welcome John "<iframe src=kVIMKTqO <" John. 	Validate
Unfiltered <iframe> src	<iframe src=oSwSpZoO></iframe>		Welcome John "<iframe src=oSwSpZoO></iframe>" John. 	Validate
Unfiltered <script> src	<script src=wLyNkQxN <		Welcome John "<script src=wLyNkQxN <" John. 	Validate



# Remediation: Web Application Firewalls

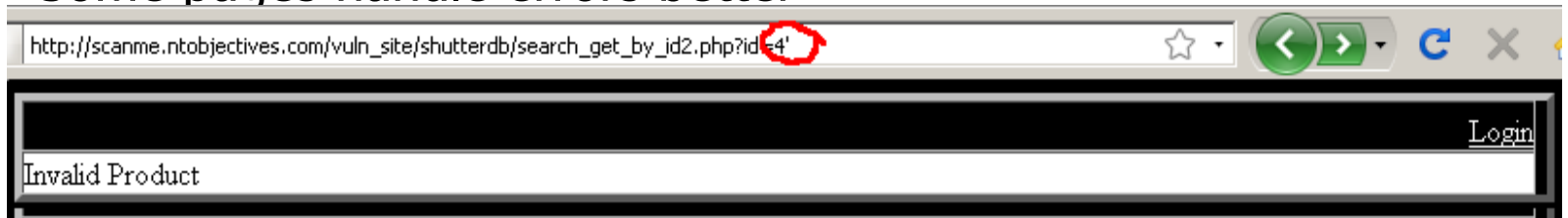
---

- Simple Installation in Environment and Quick Fix Ability
  - Since the code is not touched, security/network team can handle installation
  - Rapid Installation vs. back and forth with developers
  - Gives developers time to update the code for the proper solution
- Requires Configuration for Optimum Benefit
  - WAF must be careful not to block too early or risk breaking functionality
  - Manual configuration can be tedious on large sites
  - **Input from NTOSpider will save time and effort**

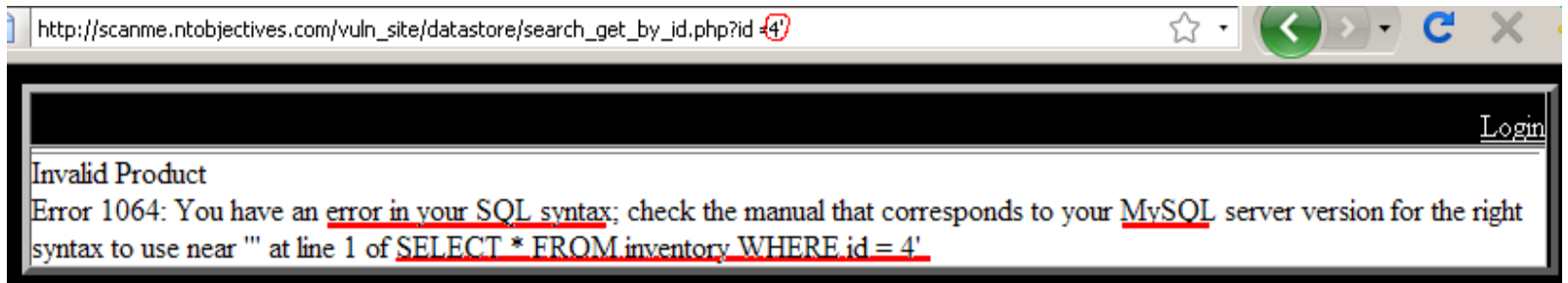


# Remediation: Make Your WAF Stronger

- Generation of pinpoint specific rules/filters for your site
  - Some pages handle errors better



- than others



- Specific rules protect weaker inputs by blocking attacks not suitable as global rules



# Remediation: PCI-DSS Before WAF & NTO

<b>Requirement 6: Develop and maintain secure systems and applications</b>		<b>Failed</b>
<b>Requirement 6.5.1: Cross-site scripting (XSS)</b>		<b>Failed</b>
URL: <a href="http://scanme.ntobjectives.com:80/vuln_site/crosstraining/">http://scanme.ntobjectives.com:80/vuln_site/crosstraining/</a>		+
URL: <a href="http://scanme.ntobjectives.com:80/vuln_site/crosstraining/aboutyou2.php">http://scanme.ntobjectives.com:80/vuln_site/crosstraining/aboutyou2.php</a>		+
URL: <a href="http://scanme.ntobjectives.com:80/vuln_site/jsmenu/z">http://scanme.ntobjectives.com:80/vuln_site/jsmenu/z</a>		+
<b>Requirement 6.5.2: Injection flaws</b>		<b>Passed</b>
Note: Primary focus on SQL and OS Command injection. Not all possible LDAP or Xpath tests are supported due to automated limitations		
<b>Requirement 6.5.3: Malicious file execution</b>		<b>Passed</b>
<b>Requirement 6.5.4: Insecure direct object references</b>		<b>Passed</b>
<b>Requirement 6.5.5: Cross-site request forgery (CSRF)</b>		<b>Passed</b>
<b>Requirement 6.5.6: Information leakage and improper error handling</b>		<b>Failed</b>
URL: <a href="http://scanme.ntobjectives.com:80/vuln_site/datastore/getimage_by_id.php?id=1">http://scanme.ntobjectives.com:80/vuln_site/datastore/getimage_by_id.php?id=1</a>		+
URL: <a href="http://scanme.ntobjectives.com:80/vuln_site/datastore/search_get_by_id.php?id=3">http://scanme.ntobjectives.com:80/vuln_site/datastore/search_get_by_id.php?id=3</a>		+
<b>Requirement 6.5.7: Broken authentication and session management</b>		<b>Passed</b>
<b>Requirement 6.5.8: Insecure cryptographic storage</b>		<b>N/A</b>
<b>Requirement 6.5.9: Insecure communications</b>		<b>Passed</b>
<b>Requirement 6.5.10: Failure to restrict URL access</b>		<b>Not Tested</b>
Note: The automated scanner is unaware of all of the locations that are intended to be restricted. Manual review of the crawled URL list and other provided data will assist the auditor in determining compliance		
<b>Requirement 6.6: For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods</b>		<b>Failed</b>
<b>Option B: Verify that a web-application firewall is in place in front of public-facing web applications to detect and prevent web-based attacks</b>		<b>Failed</b>



# Remediation: PCI-DSS After WAF & NTO

<b>Requirement 6: Develop and maintain secure systems and applications</b>		<b>Pass</b>
<b>Requirement 6.5.1: Cross-site scripting (XSS)</b>	<b>Passed</b>	
<b>Requirement 6.5.2: Injection flaws</b>	<b>Passed</b>	
Note: Primary focus on SQL and OS Command injection. Not all possible LDAP or Xpath tests are supported due to automated limitations		
<b>Requirement 6.5.3: Malicious file execution</b>	<b>Passed</b>	
<b>Requirement 6.5.4: Insecure direct object references</b>	<b>Passed</b>	
<b>Requirement 6.5.5: Cross-site request forgery (CSRF)</b>	<b>Passed</b>	
<b>Requirement 6.5.6: Information leakage and improper error handling</b>	<b>Passed</b>	
<b>Requirement 6.5.7: Broken authentication and session management</b>	<b>Passed</b>	
<b>Requirement 6.5.8: Insecure cryptographic storage</b>	<b>N/A</b>	
<b>Requirement 6.5.9: Insecure communications</b>	<b>Passed</b>	
<b>Requirement 6.5.10: Failure to restrict URL access</b>	<b>Not Tested</b>	
Note: The automated scanner is unaware of all of the locations that are intended to be restricted. Manual review of the crawled URL list and other provided data will assist the auditor in determining compliance		
<b>Requirement 6.6: For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks by either of the following methods</b>	<b>Passed</b>	
<b>Option B: Verify that a web-application firewall is in place in front of public-facing web applications to detect and prevent web-based attacks</b>	<b>Passed</b>	



NT OBJECTIVES, INC.

# Innovating Application Security

[www.ntobjectives.com](http://www.ntobjectives.com)