

Security Snake Oil
Why Known Vulnerability Checks for Web
Applications Simply Don't Work

NT OBJECTives, Inc.
White Paper
Authored by Matthew Cohen
and Dan Kuykendall

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

Security Snake Oil

Why Known Vulnerability Checks for Web Applications Simply Don't Work DRAFT

Increasing scrutiny of web application vulnerabilities by regulatory agencies and VISA/Mastercard has resulted in an effort to identify and remediate these vulnerabilities. Gartner estimates that 70% of attacks are made at the application layer. This is true for a reason as noted by Willie Sutton, "Because that's where the money is." Most web applications are custom coded and require significant expertise to protect. The application security landscape remains a target rich environment for hackers.

Some owners of web applications have turned to traditional network scanning companies to test their web applications using the same signature-based methodology used to test for layer 6 vulnerabilities. This paper will argue that this methodology is ill suited to testing web applications and will miss discovering nearly all web application vulnerabilities.

Signature-Based Checks for Layer 6 Vulnerabilities

Testing for layer 6 vulnerabilities is a well established and successful technique used broadly across the industry. Simply, a request is made to the device and a response is analyzed for a signature. If the signature exists in the response, the vulnerability exists. If not, the vulnerability does not exist. The vulnerabilities checked for have already been discovered by human beings and their exact instantiation are known at the time of the check (hence the term, "Known Vulnerability Checking").

A simple example attack against an FTP server is as follows:

```
RETR ../..../..../..../..../..../..../etc/passwd
```

Which is trying to use dot dot backslashes to request a file that should be well outside the permission space of most if not all users.

If the response matches this pattern a vulnerability has been discovered

```
root:.*:0:[01]:.*:
```

Signature-Based Checks for Layer 7 Vulnerabilities

Some security industry participants have tried to replicate the success of Layer 6 signature-based attacks to web applications. This approach is doomed to fail for several reasons.

Problem 1: Different Site Structure

Let's pick acme.com as a fictitious example of a vulnerable website. A White Hat discovers a SQL Injection vulnerability in Acme's online store:

```
http://acme.com/store/checkout.asp?creditcard='9999999999999999&expdate=1199
```

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

the single quotation mark before the credit card number would be used by a hacker to insert SQL commands instructions that could allow the hacker to view all of the credit card records.

The response from the acme.com website includes:

```
DB Error, could not query the database
MySQL Error: You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the right syntax to
use near ''9999999999999999'' at line 1
```

This attacks and response pair is added to a list of attacks that are made a part of a web application penetration test. The attack is run against another website, Mainstreet.com

The domains are swapped out and the attack is made as

```
http://mainstreet.com/store/checkout.asp?
creditcard='9999999999999999&expdate=1199
```

The first problem is that the credit card page for mainstreet.com is located at

```
http://mainstreet.com/signup/creditcard.asp
```

and there is no link

```
http://mainstreet.com/store/checkout.asp?
```

So the attack is made against a nonexistent page.

Problem 2: Different Parameter Names

The second problem is that there is no common usage of variable names across web applications.

The attack is made against the variable creditcard because that is the parameter name on the acme site.

```
creditcard='9999999999999999&expdate=1199
```

On the mainstreet.com site, the variable is called ccn so the attack would have to be

```
ccn='9999999999999999&expdate=1199
```

Here, the attack will be made against a nonexistent parameter and will fail to identify any vulnerabilities.

Problem 3: Different Responses

NT OBJECTives, Inc.

- 3 -

www.ntobjectives.com

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

A third class of problems is that web application responses vary from site to site.

If the attack had been properly classified, the response on Mainstreet.com would have been:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'  
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error near value  
'9999999999999999' on index.asp, line 5
```

Not the signature

```
DB Error, could no query the database  
MySQL Error: You have an error in your SQL syntax; check the manual  
that corresponds to your MySQL server version for the right syntax to  
use near '9999999999999999' at line 1
```

That the signature-based tool was seeking. Again, it would have missed the vulnerability.

Problem 4: Polymorphic Attacks

The above is a simplification of the difficulty of testing for web application vulnerabilities. A further, and more challenging problem, is that web application attacks are polymorphic. Each attack can be made in a large number of ways and the application must be coded to block each instantiation of the attack. In most cases, most web applications can block some or most of these attacks. Finding vulnerabilities requires a comprehensive search for these vulnerabilities.

For example, on the link above

<http://mainstreet.com/signup/creditcard.asp?visamc=9999999999999999&exp=1199>

each parameter can be attacked before and after the variable amount.

Other SQL Injection attacks include

“

Unicode attacks

CHAR

SQL Comment Delimiter

Invalid LIMIT test

Etc.

That single link could have as many as 50 x NumberOfParameters attacks just for SQL Injection. If a page has 5 parameters that's 250 SQL Injection attacks for that single page.

A recent scan by NTOBJECTives for a client site with 3,000 links ran 300,000 attacks and discovered 16 vulnerabilities. These were all specifically created for the website. None of the vulnerabilities would have been discovered by signature-based attacks.

NT OBJECTives, Inc.

- 4 -

www.ntobjectives.com

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

Problem 5: Authentication

Many web pages are only accessible if the user has properly logged into the application. Signature-based tools have no ability to log into websites and will not be able to test pages that are only available to authenticated users. For many websites, this can be a majority of pages.

The Solution: Web Application Scanners

The unfortunate solution is that web applications need to be tested either by hand or by using highly sophisticated automated tools. Signature-based checks cannot be used as they only work on the website on which the vulnerability was previously discovered by a human being. Licensing “checks” is not an option for Layer 6 toolmakers as the checks will vary for each website. Every website needs a set of checks that are custom created for it by software that has the built in artificial intelligence to accomplish this. A simple overview is appropriate.

Step 1: Crawl the Website

Web scanners are not given a finite IP and port range to outline what they need to test. They are given a start page and credentials (if applicable) and first need to discover the entire site. This is actually the most challenging part of the problem. A comprehensive discussion of the challenges of deep crawling is beyond the scope of this paper. In summary, some the difficulty is basically that web application scanners must not only support a myriad number of web technologies (e.g. JavaScript, session management, form population) but also build in intelligence to walk through the site so that all of the pages are crawled and later tested. If a page is not crawled, it will not be assessed.

Step 2: Understand Normal Website Responses

Web scanners must also understand normal application responses so that they can know when a vulnerability exists. Again, this cannot be signature-based. Web scanners must have advanced heuristics to mimic the human brain's ability to distinguish subtle differences between pages.

Step 3: Craft Custom Attacks

Once the application has been inventoried and analyzed, the web scanner must craft custom attacks that target the site's specific architecture. These will vary from site to site as illustrated above.

Step 4: Make Attacks

The scanner then fires off tens or hundreds of thousands of attacks and sees where the site is vulnerable.

Step 5: Report Vulnerabilities

A report is generated showing the enterprise where the site is vulnerable and consolidating the vulnerabilities for easy remediation.

Conclusion

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

A broad understanding is gradually penetrating the security field: signature-based tests, like those offered by the \$100 PCI firms, are nothing more than a false sense of security. The utter failure of some of these tools to identify obvious vulnerabilities in public websites has been well publicized. Signature-based web application tests have little or no utility unless you are running the identical version of the application that has been tested thoroughly by hand (and all vulnerabilities have been placed in the signature-based testing tool). This is almost never the case. Signature-based web application testing is, in effect, the subprime mortgage of the security world.

Buyers of both software and service-based VA solutions are increasingly demanding a complete and integrated solution that adequately tests their network for security risks. This is true to satisfy regulators, PCI requirements as well as internally created security standards. This solution includes both signature-based tests for Layer 6 security and industrial strength AI-based tests for Layer 7.

Network security vendors have started to add web application tests to their testing portfolio. Some have leveraged the familiar signature-based methodology, which this paper has argued is doomed to fail. Some have tried to replicate the 20-50 man years of effort required to build a proper AI-based web scanner with a fraction of the investment. And some network VA toolmakers have chosen to partner with established players with proven tools. As market (and regulatory) awareness of the requirements of web application scanning increase, buyers are increasingly able to distinguish between tools based on their ability to accurately assess their exposure to web application attacks.

About the Authors

Matthew Cohen is the Chief Operating Officer of NT OBJECTives, Inc.

Mr. Cohen brings extensive experience in the investment banking industry and executive management to NTO, where he manages the organization's operations and finances. Previous to NTO, Matthew was CFO of publicly traded TTR Technologies. Prior to TTR, Mr. Cohen was the founding CFO at APB Online, Inc., where he raised \$27 million in three private placements and built a financial infrastructure to support a 140 person company. Matthew has held positions at The Blackstone Group, Rothschild, Inc., and Kidder, Peabody & Co.; where he worked on restructurings, capital financings, and mergers and acquisitions. In one of these restructurings, New Dartmouth Bank, the investor group acquired 3 insolvent banks from the FDIC, invested \$40 million to re-capitalize them and after an operating restructuring, sold the bank for \$160 million two and a half years later. He holds a degree in economics from Princeton University.

Dan Kuykendall is the Director of Engineering of NT OBJECTives, Inc.

Responsible for driving NT OBJECTives' research and development efforts. Mr. Kuykendall brings an extensive background in web application development methodologies and security related understanding to NT OBJECTives. Dan joins NT OBJECTives from Foundstone, where he was responsible for the web interface to the companies flagship product, FoundScan. During this time he was instrumental in building scan management, and remediation capabilities into the product. Prior to Foundstone,

NT OBJECTives, Inc.

- 6 -

www.ntobjectives.com

Security Snake Oil: Why Known Vulnerability Checks for Web Applications Simply Don't Work

Dan led the foundation of the Information Security team in the United States branches of the financial giant, Fortis. Mr. Kuykendall is involved with Web Application Security Consortium, is regular contributor to many open source development projects, and podcasts to educate the public about web application security issues.