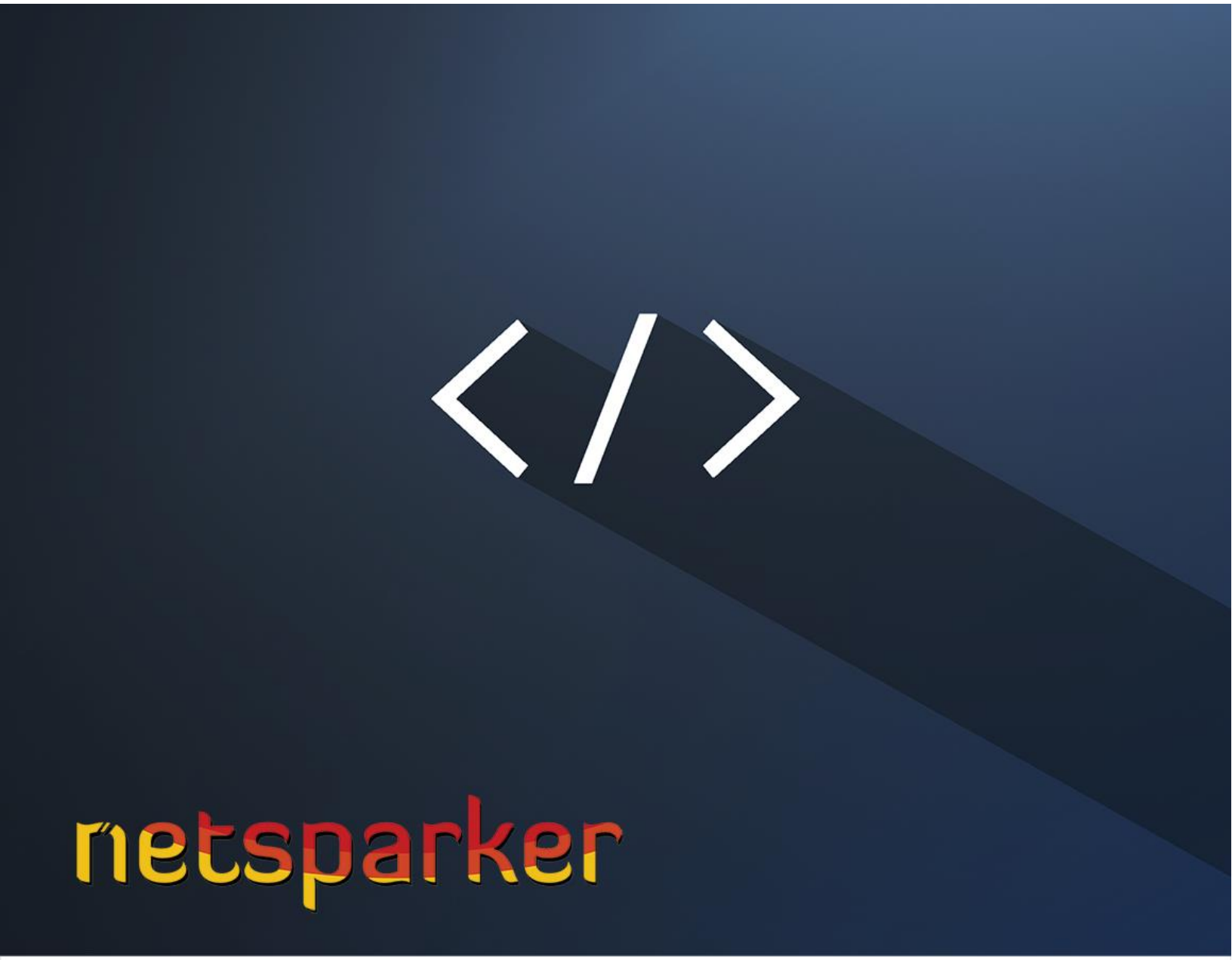


Automating the Configuration of URL Rewrite Rules in Netsparker Web Application Security Scanners

Allowing you To scan more websites without the need of getting bogged down into configuring the scanners

Robert Abela

January 2016

Executive Summary

Modern websites and web applications make use of the URL rewrite technology for search engine friendly and human readable URLs. But URL rewrite technology also opened new security holes and made it difficult for web application security scanners to automatically detect vulnerabilities.

The automated web application security scanners vendors responded to the situation by introducing new functionality that allows users to configure their tools and still be able to automatically scan websites which use URL rewrite technology. Though configuring automated tools resulted in being difficult, hence many users do not configure their scanners, potentially leaving a good number of security flaws unnoticed.

At Netsparker, we believe that a web application security scanner should be easy to use so you can scan as much web applications as you can in the shortest time possible. After all, that is the whole point of automation, no? Imagine you have to scan 100, or even worse, 1000 websites. How long will it take you to complete such task? First you have to find the developers and get all the information from them, which is no easy feat, especially in big companies. Then you have to manually configure the scanner. Quite a laborious process, isn't it?

Therefore, we have developed a new heuristic technology that can automatically determine the setup of the target web application and configure the scanner without the need for any user interaction.

This whitepaper looks into the shortcomings of existing automated web application security scanners, and explains in detail the new heuristic URL rewrite technology in both Netsparker Desktop and Netsparker Cloud web application security scanners.

What is URL Rewrite?

URL rewrite is a web server technology that is used to rewrite URLs into *human readable* URLs. Therefore, if for example you want to access a blog post titled **first-post** you have to use the following URL since modern web applications such as blogs store data in backend databases:

http://www.example.com/blog/show.php?post=first-post

Though the above URL is not easy to neither remember nor is it search engine friendly. So by configuring URL rewrite technology on your web server, the web server can accept friendly URLs such as the below and translate them for the web application to understand them:

http://www.example.com/blog/first-post/

From the example above we can see that the page *show.php* has a parameter **post** and when the parameter value is **first-post** the web application retrieves a post titled **first-post** from the backend database. For more information on this subject you can also read Wikipedia's entry on Rewrite Engine.

Automated Web Security Scanners' Shortcomings

In theory, the way an automated web security scanner works is very simple. First it crawls the website to identify all possible attack surfaces, such as parameters that accept user or dynamic input, and then it attacks these parameters. Though things are not that simple when scanning a website that has URL rewrite enabled. As explained in the example above since the parameter is *invisible* the automated scanner won't find it unless it is configured to do so, hence such parameters are not scanned for vulnerabilities, and the scans will take much longer to finish.

In some cases, it is impossible to scan a website that uses URL rewrite rules unless you configure the scanner. This can happen if the target website has both URL rewrite rules and custom 404 error pages configured, and the scanner starts requesting non-existing URLs. Such scenario is very common with scanners that have an extensive list of backup file checks, i.e. they try to identify possible backup files on the target web application.

Manually Configuring URL Rewrite Rules in Web Security Scanners

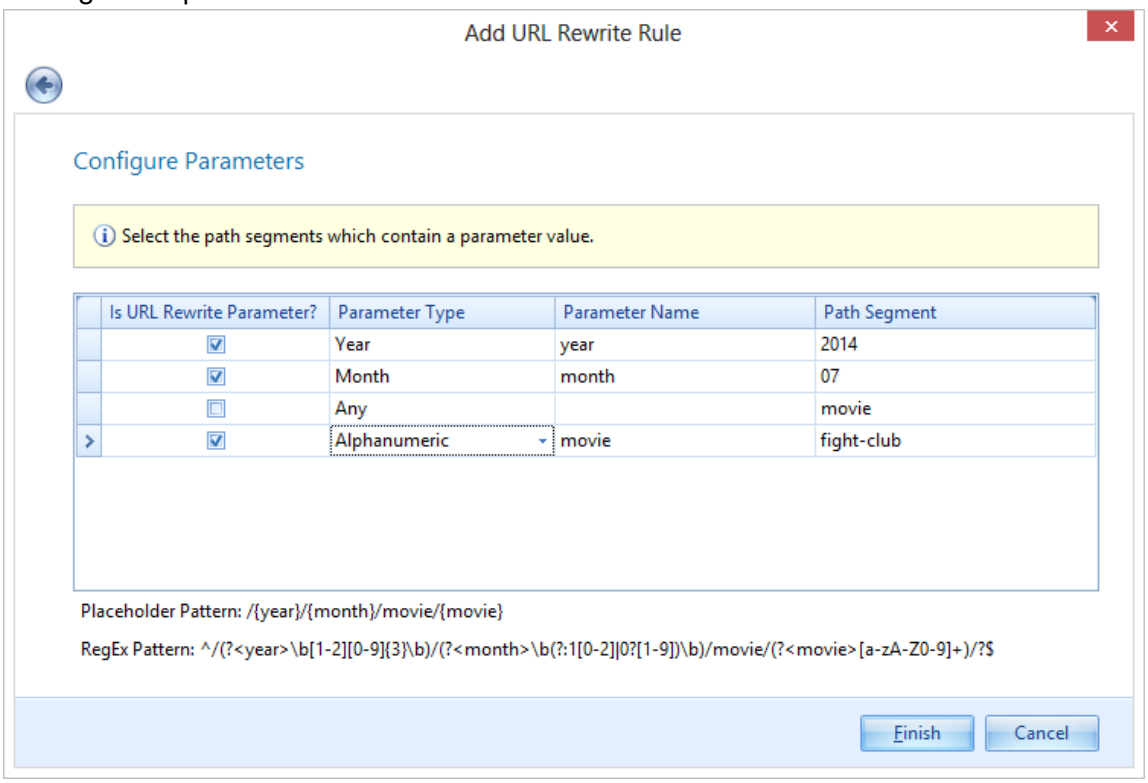
Once URL rewrite rules are configured in an automated web application security scanner, it will crawl the website more efficiently, and it will identify and scan all the input parameter, including those used in URLs. The duration of the security scan will be much less as well.

Even though there are several advantages to configuring URL rewrite rules in web security scanners, through our support and sales we have learnt that in most cases users still do not configure them. The most common reasons for not configuring the scanners are:

- Users do not have enough knowledge about the target website,
- Users are unable to write regular expressions which are needed when configuring URL rewrite rules in scanners,
- The process is very difficult and time-consuming.

Easy to Configure URL Rewrite Rules

After learning about these pain points, we developed a wizard that helps users configure URL rewrite rules in web security scanners within just a few minutes, without the need to know how to write regular expressions.



The end result was a very simple two-step process;

- Specify a sample URL
- Specify the parameter name and type

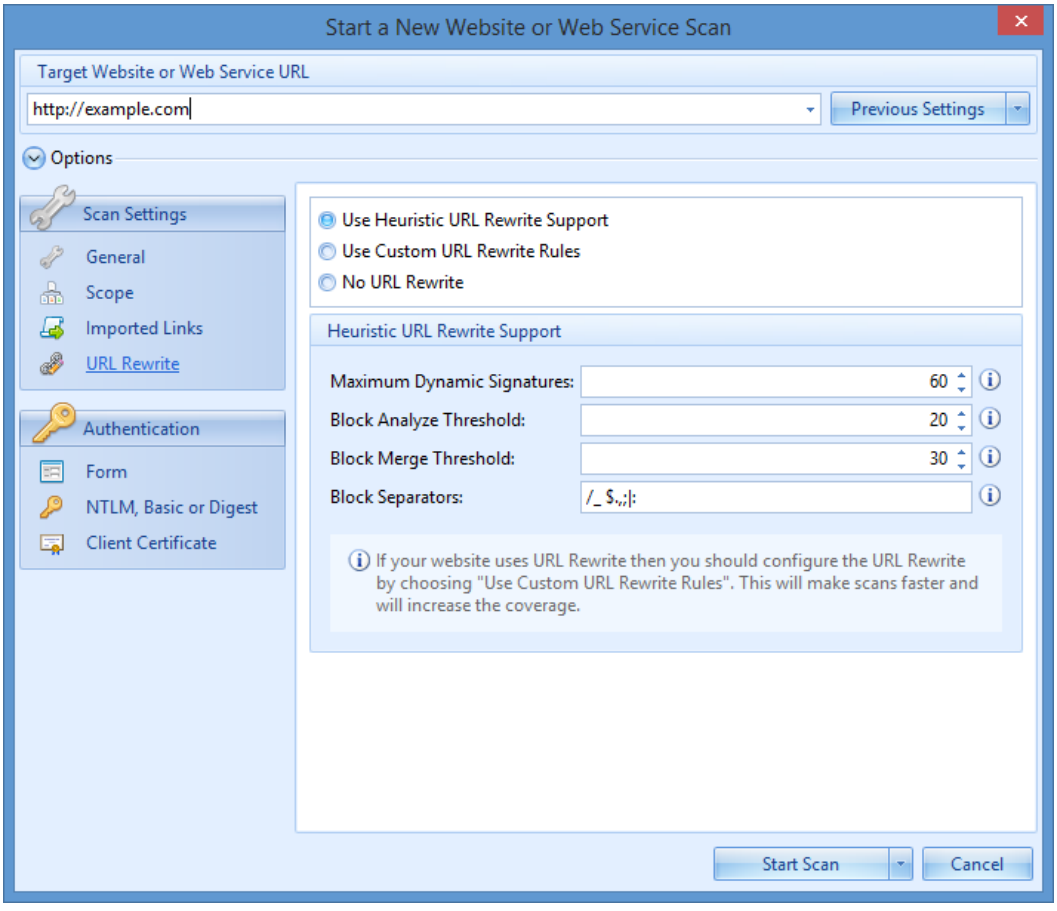
The rest is all done automatically by the scanner as shown in [How to Configure URL Rewrite Rules in Netsparker].

Automating URL Rewrite Configuration in Netsparker Web Application Security Scanners

Even though we simplified the configuration of URL rewrite rules in our scanners, the solution still had a shortcoming; the user needed to have knowledge about the target website to be able to specify the parameter(s) name. Therefore, we opted to proceed with a fool proof solution; automate the whole process so users can scan websites which use URL rewrite technology without the need to configure anything.

How Does it Work?

The concept behind the technology is very easy to understand. Once you launch a security scan, the Netsparker web application security scanner will start crawling the website. During this stage, every time the scanner crawls 20 URLs it will check if 60 of the crawled URLs have a matching URL pattern. Even though the above mentioned values apply for most of the cases they are also are configurable, as shown in the below screenshot.



How Does Netsparker Web Security Scanner Match the Patterns of URLs?

The scanner splits every crawled URL into blocks. For example the URL *http://www.example.com/blog/first-post/* is split in the following blocks:

Block 1: /
Block 2: blog
Block 3: /
Block 4: first-post
Block 5: /

After crawling enough URLs, the scanner is able to notice that the all the blocks in the crawled URLs are the same except for Block 4. Therefore, Block 4 is a dynamic parameter and the scanner automatically configures the URL rewrite rules to be able to efficiently crawl the website and attack the parameters.

Heuristic Technology Can Identify Multiple Parameters in URLs

Netsparker's new heuristic URL rewrite engine has more advanced checks so it can also automatically configure URL rewrite rules in more advanced setups, such as when websites use multiple parameters in a URL. Imagine a website which has the following URL structure:

> *http://www.example.com/category1/page1 -> /category1/page30*
>
> *http://www.example.com/category2/page31 -> /category2/page60*

For a human, it is easy to realize that two parameters are being used in these type of URLs. A parameter is being used for the category and a parameter for the page. But it is not so obvious for an automated web security scanner. In such cases, this is how Netsparker web application security scanners manage to automatically identify all the parameters in the URLs:

1. The scanner identifies 60 matching URLs and creates the first URL rewrite rule

Even though the parameter value in the first block of the URL is different (category1 and category2), the 60 URLs in the above example share the same pattern:

http://www.example.com/category[number]/page[number]

Hence, the scanner creates the first URL rewrite rule to scan the *page* parameter, something like; /category1/{param1}

2. If during the scan the scanner crawls a number of URLs that in them there are more than 60 different categories, then it will create another URL rewrite rule to scan the category parameter.

For example the scanner crawls the following URL:

http://www.example.com/category60/page1823

To crawl to this URL, the scanner crawled a number of URLs that had *category1*, *category2*, *category3* and so on in them. At this stage, the scanner updates the URL rewrite rule to be able to identify and scan both parameters in the URL; /{param1}/{param2}.

How Does the Scanner Determine Which Parts of the URLs are Blocks?

By default, the scanner has a number of pre-configured characters that are called *block separators*. The list includes the following characters: / **_ $. , ; | :**

Thanks to this list of *block separators* the scanner can easily split and analyse URLs. Like all other heuristic URL rewrite technology settings, the list of *block separators* is configurable. For more information on all the configurable options refer to the section **Fine Tuning the Automatic Detection and Configuration of URL Rewrite Rules** in Automatic Configuration of URL Rewrite Rules in Netsparker Web Application Security Scanners.

Statistics and Test Results

Netsparker's new heuristic URL rewrite technology has been extensively tested in several different scenarios. This section includes statistics about one of the simplest tests we have done to highlight the functionality and effectiveness of this new technology.

The Target Website

The target website that we will be using in this example is an ASP.NET blog web application and runs on IIS 7. Here is more information about the target website:

URL rewrite configuration:

```
routes.MapPageRoute(
        "blog-details",
        "blog/{*blogId}",
        "~/Blog.aspx"
        );
```

URL rewrite structure: *http://www.example.com/blog/{dynamic_parameter}*

Number of Blog posts: 95

Sample URLs:

http://www.example.com/blog/is-bitcoin-really-used-by-people-1/
http://www.example.com/blog/what-are-the-advantages-of-bitcoin-2/
...
...
http://www.example.com/blog/is-bitcoin-fully-virtual-and-immaterial-94/
http://www.example.com/blog/is-bitcoin-anonymous-95/

Note: All the blog post titles (the parameter values) consists of letters, numbers and hyphen.

Vulnerability: The parameter **title**, which accepts blog posts' title as value is vulnerable to boolean SQL injection vulnerability.

How Did We Conduct this Test?

The Netsparker scanning engine has three URL rewrite settings:

- No URL Rewrite Support
- Custom URL Rewrite Support
- Heuristic URL Rewrite Support

Note: The new heuristic URL rewrite technology featured in this whitepaper is replacing the old heuristic technology, which was being used to limit the scan in case URL rewrite technology was being used on the target web application.

For this test, we scanned the target website with Netsparker web application security scanner configured in all the different modes and compared the scanned results. Let's see all of the above theory in action.

No URL Rewrite Support

When you select this option, the Netsparker scanning engine presumes that the target website does not have URL rewrite configured. When scanning the target website in this mode, Netsparker:

- crawled all 95 blog posts
- did not identify the parameters in the URL (expected)
- did not find the SQL injection in the parameter (expected)
- sent 25,867 HTTP requests
- the scan took 35 minutes to complete.

Conclusion: It all worked as expected. The parameter in the URL was not crawled and the vulnerability was not identified.

Old version of Heuristic URL Rewrite

With the old version of the heuristic URL rewrite technology, the Netsparker scanning engine was just able to determine if a target website was using URL rewrite or not, so if it were it would limit the scan to not to end up in some sort of loop. This limitation of this technology was that it only recognized numeric parameters, hence it was very limited. Nonetheless, when scanning the target website in this mode, Netsparker:

- crawled all 95 blog posts
- did not identify the parameter in the URL (expected)
- did not find the SQL injection in the parameter (expected)
- sent 25,584 HTTP requests
- the scan took 35 minutes to complete

Conclusion: As expected the old heuristic URL rewrite technology did not recognize the parameter values since they are not numeric values. Also, the parameter in the URL was not crawled, and the vulnerability was not identified.

Custom URL Rewrite Support

This is the most efficient mode of them all, but it requires the user to have knowledge about the target website (hence why we developed the new automated technology). When you select this

option, you have to use the wizard to configure the URL rewrite rules in Netsparker. Therefore in this mode the scanner knows exactly what it is going to scan.

Before we launched the scan, we configured the following URL rewrite rules in Netsparker:

URL Rewrite Format : /blog/{title}/
URL RegEx/Custom Format: ^/blog/(?<title>[^/]+)/?$

And these are the scan statistics:

- scanner crawled nine blog posts only (ignored 86 posts)
- the parameter **title** was successfully crawled
- the SQL injection in **title** was identified and reported
- the scanner sent 20,650 HTTP requests
- the scan Scan took 20 minutes to complete

Conclusion: The scanner crawled the parameter and identified the vulnerability. It is important to highlight that:

- the scanner sent fewer requests since it did not need to crawl all the pages
- the scan consumed less bandwidth and resources
- the scan was around 45% faster than the previous scans

Note: Since the scanner had the URL rewrite rules manually configured one would expect it to crawl only one page and not nine pages as in the above scenario. In this case, the scanner still crawls nine pages since it is configured to do so in the crawling setting **Maximum Signatures**, which can be configured in the **Scan Policy Editor**. The scope of this setting is used as a confirmation, to ensure the type of URLs it needs to scan with the URL rewrite rules.

New Automated Heuristic URL Rewrite Support

There are a number of settings you can configure in the new automated Heuristic URL Rewrite support options but the defaults work well in most cases. The options should only be tweaked in edge cases, if you really need to. So to scan a website that uses URL rewrite just select this option and let the scanner do the work. Let's see how this mode fares when compared to the others. Here are the statistics:

- the scanner crawled 60 pages (ignored 35)
- the scanner configured its own URL rewrite rule
    - URL Rewrite format: /blog/{param1}
    - URL RegEx: ^/blog/(?<param1>[^/_\s$.,;|:]+)/?$
- the parameter **title** was successfully crawled
- the SQL injection in **title** was identified and reported
- the scanner sent 25,000 HTTP requests
- the scan took 25 minutes to finish

Conclusion: The scanner crawled the parameter and identified the vulnerability automatically. When compared to the scan with the manually configured URL rewrite rules, there are some important things worth pointing out:

- In this scan, the scanner crawled more blog posts. This is expected since it needed to crawl at least 60 URLs before it can sample the data.
- Because it needed to sample 60 URLs, it sent more HTTP requests and the scan was 25% longer than when the rewrite rules were manually configured.

The Benefits of the New Heuristic URL Rewrite Rules Technology

The new heuristic URL rewrite technology clearly works and brings along a good number of benefits to the table, such as:

Ease of Use: this new heuristic technology allows users who do not have enough knowledge of the target web application to still be able to scan it automatically.

Better Scan Coverage: if no URL rewrite rules are configured in the scanner when scanning a website that uses URL rewrite technology, the scanner will fail to crawl several parameters and will not scan them.

More Accurate Scan Results: this new automated technology will make automated web vulnerability scans less prone to common human errors, which typically lead to incorrect scan results.

Scaling Up is Easy: Imagine you have to scan 1000 websites with Netsparker Cloud. Can you get all the information about the target websites and configure the scanner in a timely manner? No, and man hours are far more expensive than automation in a scanner.