Imperva WAF Gateway – Tuning Web Profiles

Jaired Anderson, Sales Engineer

Web Profiles are a feature of the Imperva WAF Gateway platform whereby dynamic profiling technology automatically examines web application traffic to create a comprehensive profile of its structure and behavior.

As WAF Gateway resources are finite, there are limits to the number of data points the profiling engine will learn by default.

Some learned web profiles may require tuning to stay within the defined thresholds while others won't; it's entirely dependent upon the structure of the web application.

The default web profiling engine thresholds are defined in the table below.

Property	Limit
Maximum allowed hosts per application	4000
Maximum allowed URLs per directory	500
Maximum allowed parameters per URL	200
Maximum allowed parameters per application	3000
Maximum allowed URLs per application	5000
Maximum allowed sub-directories per directory	500
Maximum allowed parameters per URL pattern	200
Maximum allowed parameters per SOAP action	300
Maximum allowed SOAP actions per URL	100
Maximum allowed cookies per application	200
Maximum allowed action URLs per application	200
Maximum allowed login URLs per application	200

The above thresholds can be increased from the default values (at the discretion of Imperva support) but also another alternative , in many cases is a simple Web Profile "tune up".

This blog covers two of the most common scenarios that cause one to exceed their profile size limits and how to resolve them.

We'll start by accessing the profile under **Main > Profile > Overview**

The hazard icon below indicates the presence of an optimization issues.

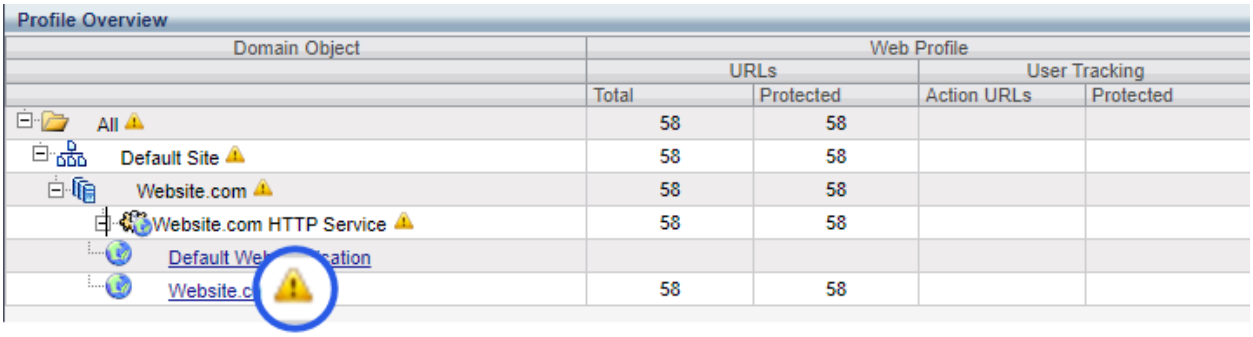


Figure 1

Click the hazard icon to access the **Optimization** screen (1), which will provide a short summary of the issues (2) along with recommendations.(3)

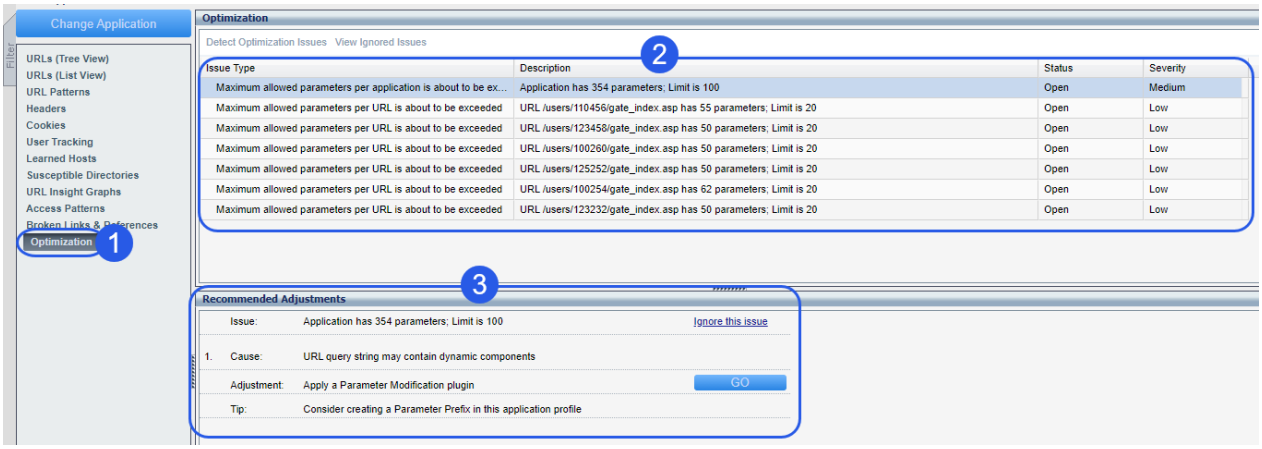


Figure 2

Note: For the purposes of this blog, I have intentionally lowered the threshold limits to trigger optimization issues.

Let's review the top two recurring issues.

- Application has 354 parameters; Limit is 100
- URL "x" has "x" parameters; Limit is 20

Since we understand the optimization issues, let'srevisit the web profile structure by clicking **URLs (List View)** from the menu. Click the red arrow twice next to **# of Params** to sort in descending order. We are looking for URLs that will provide the most return with the least tuning effort. In this case, the top 6 URLs contain 62, 55, 50, 50, 50, and 50 parameters each and are a candidate for tuning. Likewise, the remaining URLs contain 5 parameters or less, thuswould not be suitable candidates for tuning.

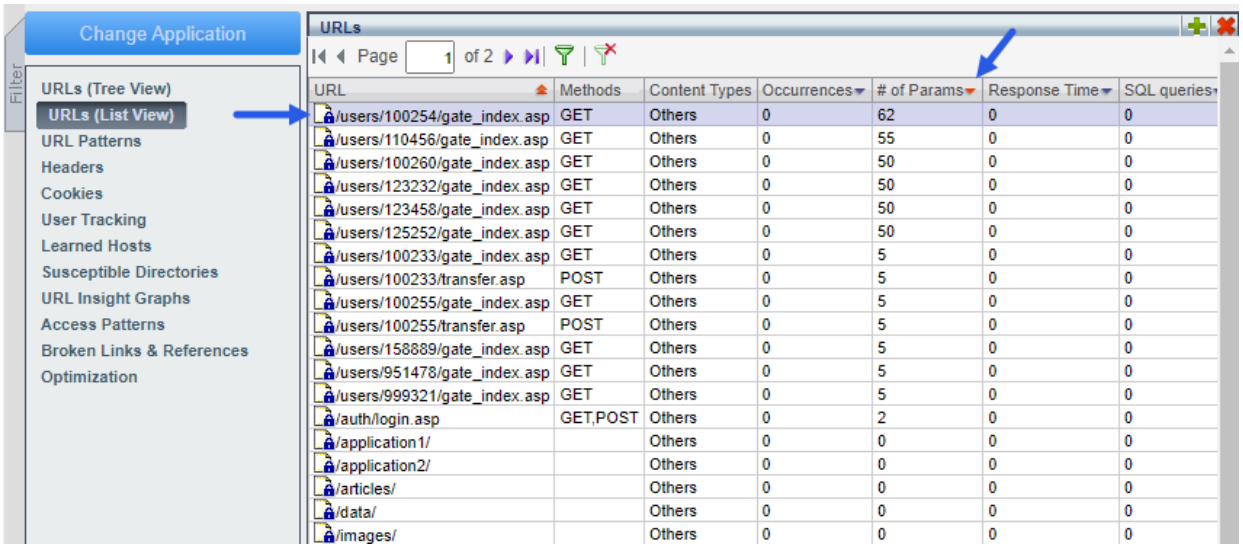


Figure 3

Selecting a URL from the list will allow us to review the parameters associated with that URL in the right hand pane. (Fig 4) While reviewing the parameters, we can identify several patterns. For example, section (1) contains two address fields. The pattern that we recognize is "gate_index-address-" a constant value, is followed by a numeric variable. We see the same behavior in sections (2) and (3). However, following the "best bang for the buck" model – these entries would not be desirable for tuning efforts. Section (4) is a prime candidate for tuning as we see 50 fields matching a pattern and the Min/Max value is also the same for each parameter.

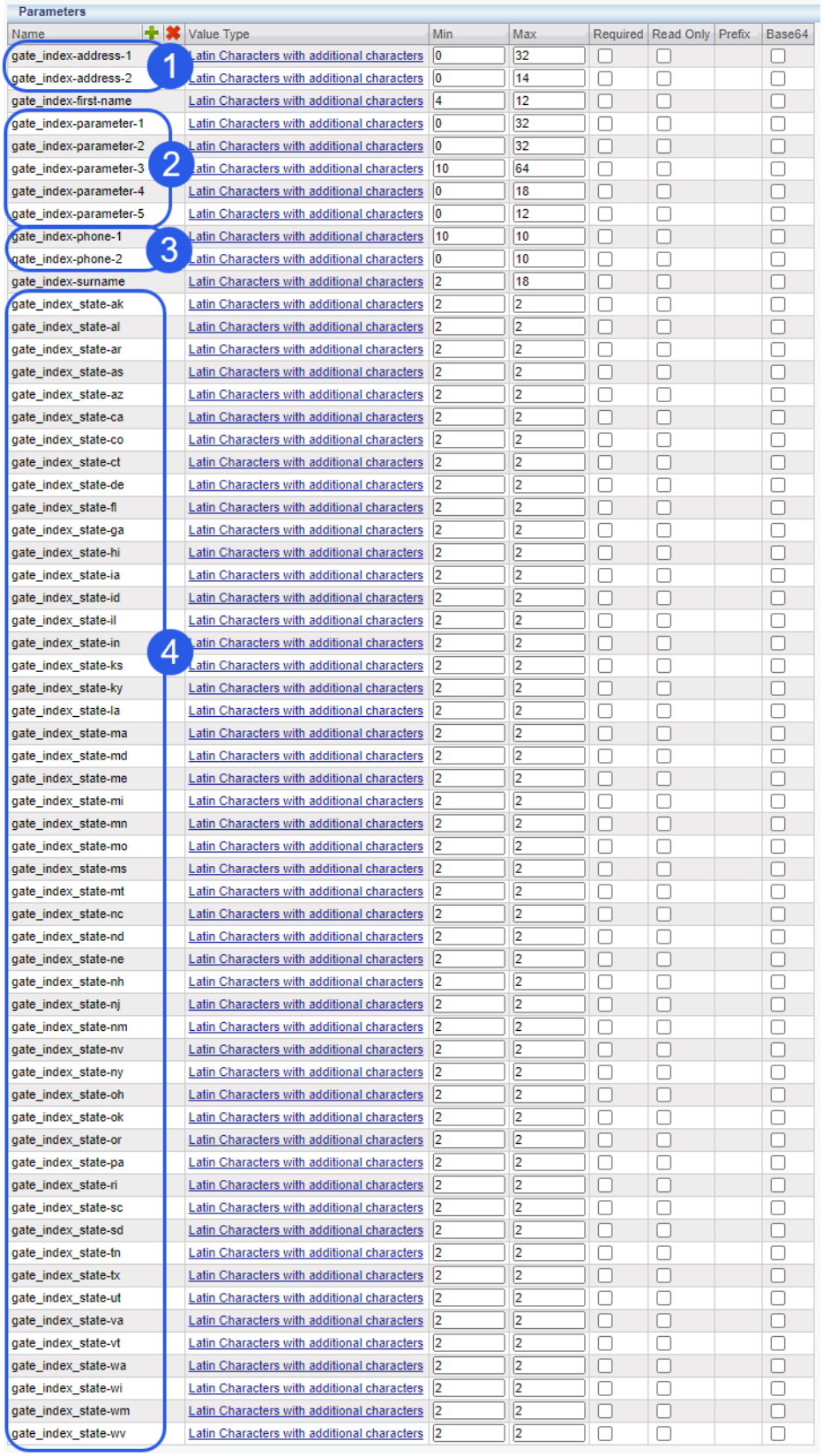
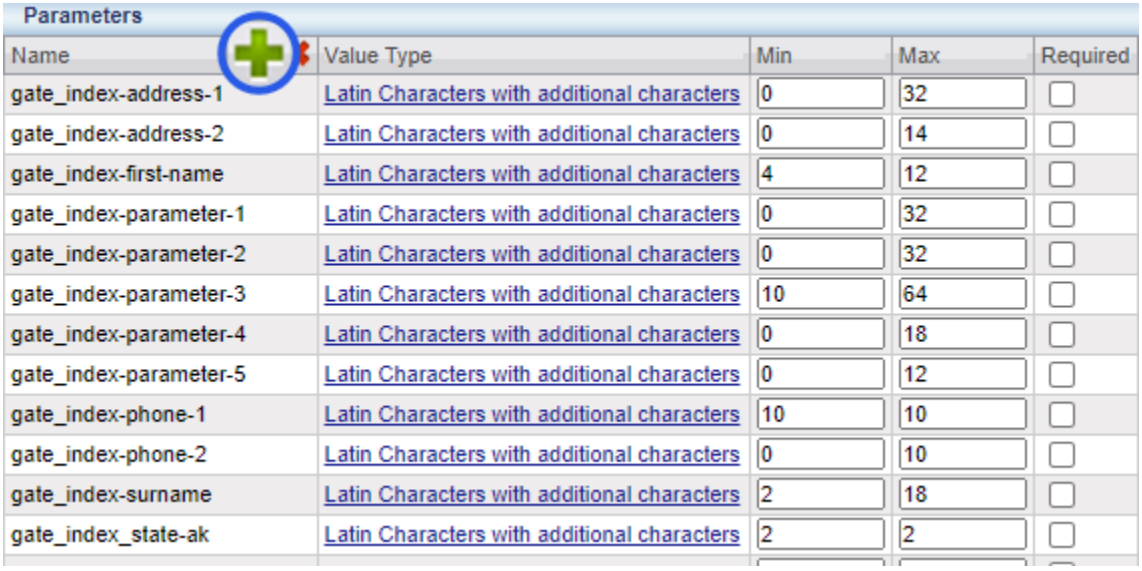

Parameters
Name
Value Type
Min
Max
Required
Read Only
Prefix
Base64
gate_index-address-1
gate_index-address-2
gate_index-first-name
gate_index-parameter-1
gate_index-parameter-2
gate_index-parameter-3
gate_index-parameter-4
gate_index-parameter-5
gate_index-phone-1
gate_index-phone-2
gate_index-surname
gate_index_state-ak
gate_index_state-al
gate_index_state-ar
gate_index_state-as
gate_index_state-az
gate_index_state-ca
gate_index_state-co
gate_index_state-ct
gate_index_state-de
gate_index_state-fl
gate_index_state-ga
gate_index_state-hi
gate_index_state-ia
gate_index_state-id
gate_index_state-il
gate_index_state-in
gate_index_state-ks
gate_index_state-ky
gate_index_state-la
gate_index_state-ma
gate_index_state-md
gate_index_state-me
gate_index_state-mi
gate_index_state-mn
gate_index_state-mo
gate_index_state-ms
gate_index_state-mt
gate_index_state-nc
gate_index_state-nd
gate_index_state-ne
gate_index_state-nh
gate_index_state-nj
gate_index_state-nm
gate_index_state-nv
gate_index_state-ny
gate_index_state-oh
gate_index_state-ok
gate_index_state-or
gate_index_state-pa
gate_index_state-ri
gate_index_state-sc
gate_index_state-sd
gate_index_state-tn
gate_index_state-tx
gate_index_state-ut
gate_index_state-va
gate_index_state-vt
gate_index_state-wa
gate_index_state-wi
gate_index_state-wm
gate_index_state-wv
Latin Characters with additional characters
1
2
3
4

Figure 4

We will now define the suitable parameter as a prefix. Click the green + to add a new entry.

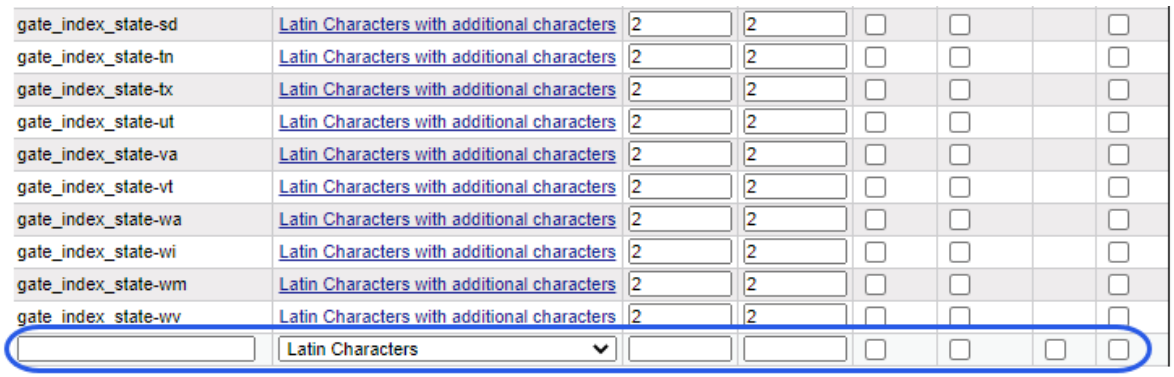


Parameters

Name	Value Type	Min	Max	Required
gate_index-address-1	Latin Characters with additional characters	0	32	
gate_index-address-2	Latin Characters with additional characters	0	14	
gate_index-first-name	Latin Characters with additional characters	4	12	
gate_index-parameter-1	Latin Characters with additional characters	0	32	
gate_index-parameter-2	Latin Characters with additional characters	0	32	
gate_index-parameter-3	Latin Characters with additional characters	10	64	
gate_index-parameter-4	Latin Characters with additional characters	0	18	
gate_index-parameter-5	Latin Characters with additional characters	0	12	
gate_index-phone-1	Latin Characters with additional characters	10	10	
gate_index-phone-2	Latin Characters with additional characters	0	10	
gate_index-surname	Latin Characters with additional characters	2	18	
gate_index_state-ak	Latin Characters with additional characters	2	2	

Figure 5

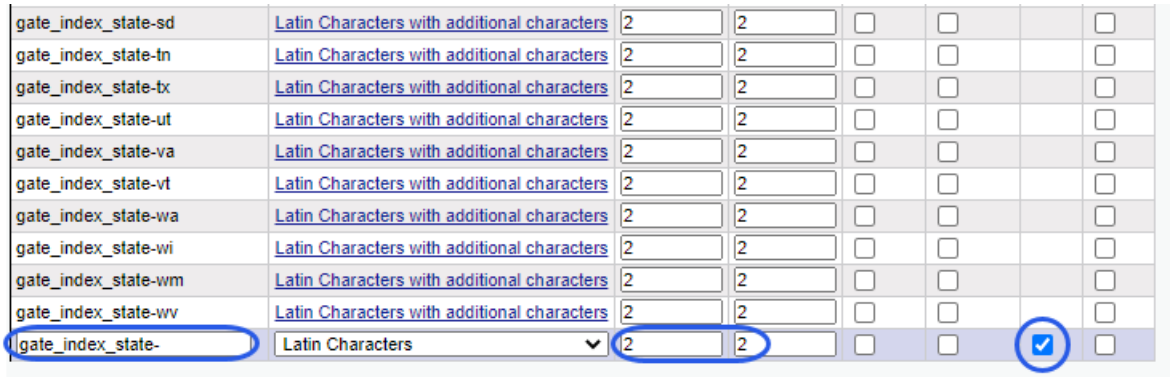
The new entry will be added to the bottom of the list, so we will need to scroll down.



gate_index_state-sd	Latin Characters with additional characters	2	2				
gate_index_state-tn	Latin Characters with additional characters	2	2				
gate_index_state-tx	Latin Characters with additional characters	2	2				
gate_index_state-ut	Latin Characters with additional characters	2	2				
gate_index_state-va	Latin Characters with additional characters	2	2				
gate_index_state-vt	Latin Characters with additional characters	2	2				
gate_index_state-wa	Latin Characters with additional characters	2	2				
gate_index_state-wi	Latin Characters with additional characters	2	2				
gate_index_state-wm	Latin Characters with additional characters	2	2				
gate_index_state-wv	Latin Characters with additional characters	2	2				
	Latin Characters						

Figure 6

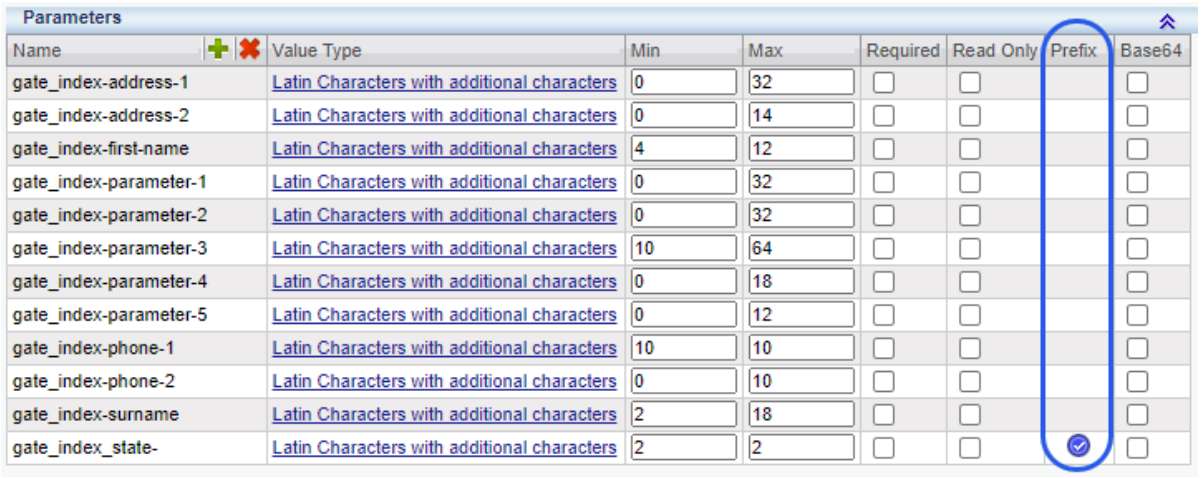
In the name field, enter the portion of the variable that is static; in this case it will be "gate_index_state-". Enter 2 in the min/max field. Place a check in the third box to indicate this is a prefix. (Fig 8)



gate_index_state-sd	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-tn	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-tx	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-ut	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-va	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-vt	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-wa	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-wi	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-wm	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-wv	Latin Characters with additional characters	2	2	☐	☐		☐
gate_index_state-	Latin Characters	2	2	☐	☐	☑	☐

Figure 7

Click **Save** at the top right. The parameter names that match the prefix will collapse to a single parameter and a check mark is present in the prefix column to indicate this entry is a prefix. We have just reduced 50 parameters to 1 parameter with a variable name.



Parameters

Name	Value Type	Min	Max	Required	Read Only	Prefix	Base64
gate_index-address-1	Latin Characters with additional characters	0	32	☐	☐		☐
gate_index-address-2	Latin Characters with additional characters	0	14	☐	☐		☐
gate_index-first-name	Latin Characters with additional characters	4	12	☐	☐		☐
gate_index-parameter-1	Latin Characters with additional characters	0	32	☐	☐		☐
gate_index-parameter-2	Latin Characters with additional characters	0	32	☐	☐		☐
gate_index-parameter-3	Latin Characters with additional characters	10	64	☐	☐		☐
gate_index-parameter-4	Latin Characters with additional characters	0	18	☐	☐		☐
gate_index-parameter-5	Latin Characters with additional characters	0	12	☐	☐		☐
gate_index-phone-1	Latin Characters with additional characters	10	10	☐	☐		☐
gate_index-phone-2	Latin Characters with additional characters	0	10	☐	☐		☐
gate_index-surname	Latin Characters with additional characters	2	18	☐	☐		☐
gate_index_state-	Latin Characters with additional characters	2	2	☐	☐	☑	☐

Figure 8

To see the new URL count we will need to refresh the screen. We can now see the parameter count for /users/100254/gate_index.asp has dropped from 62 parameters to 12. (Figure 9) Additionally, we can see there are five other pages that could benefit from the creation of a parameter prefix. We could create these manually since it is only five pages; but would one do withhundreds, or even thousands of these pages?

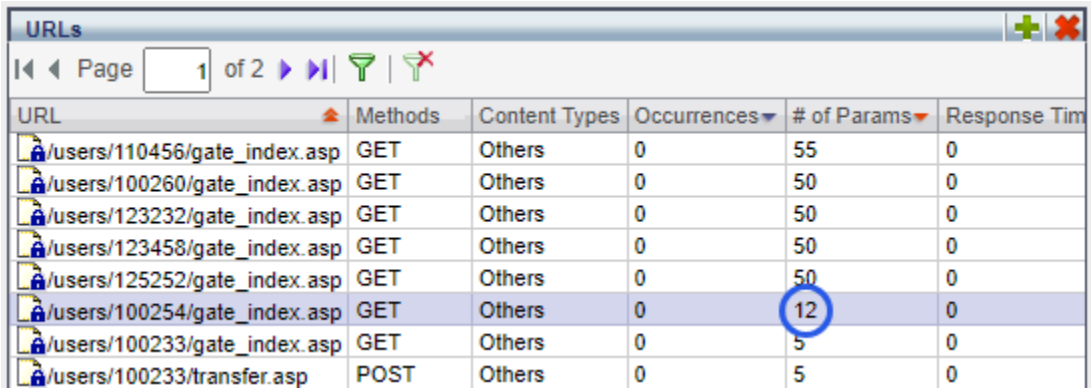


Figure 9

We've reviewed the parameters associated with the gate_index.asp and are confident the page functionality is fairly consistent across paths. In other words, this is really the same page over and over, perhaps with some minor parameter changes. However, the WAF Gateway treats these as unique pages since the path is changing. (eg: /users/**<6 numeric digits>**/gate_index.asp)

Since the page name "gate_index.asp" is unique to this particular path within the application, we can create a suffix to instruct the WAF Gateway to treat it as a single page with a variable path.

Right click on the desired URL and select **Save as Pattern**. (Figure 10)

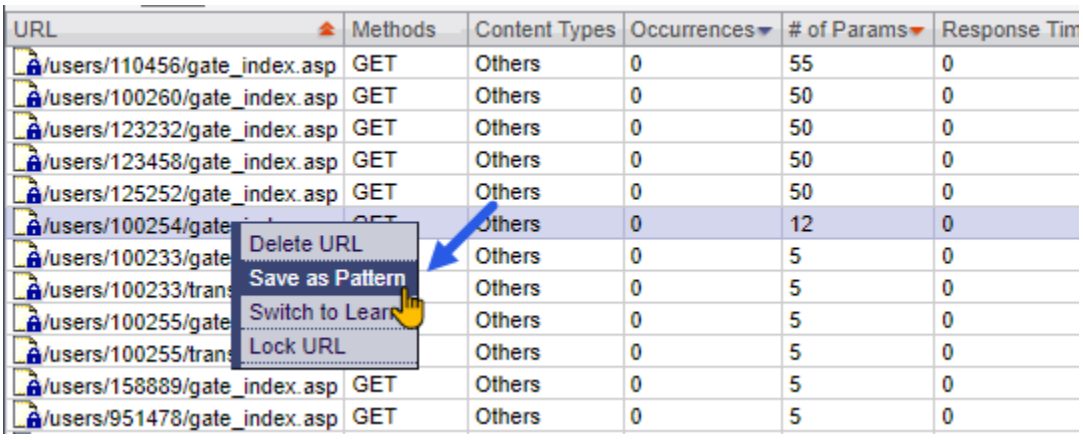


Figure 10

In the *URL Pattern* field, reduce the URL pattern to the page name excluding the path. In this case it will be "/gate_index.asp". Change the *Type* to Suffix and click **Save**.

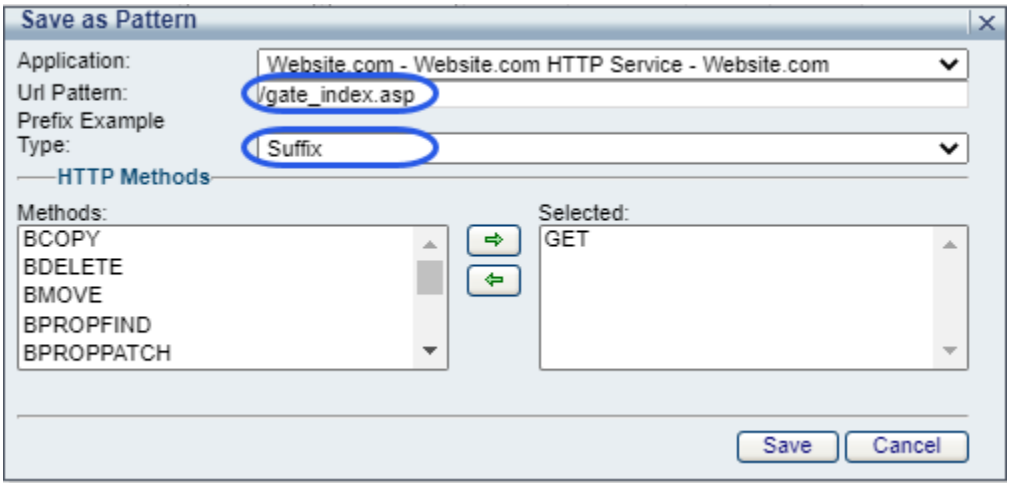
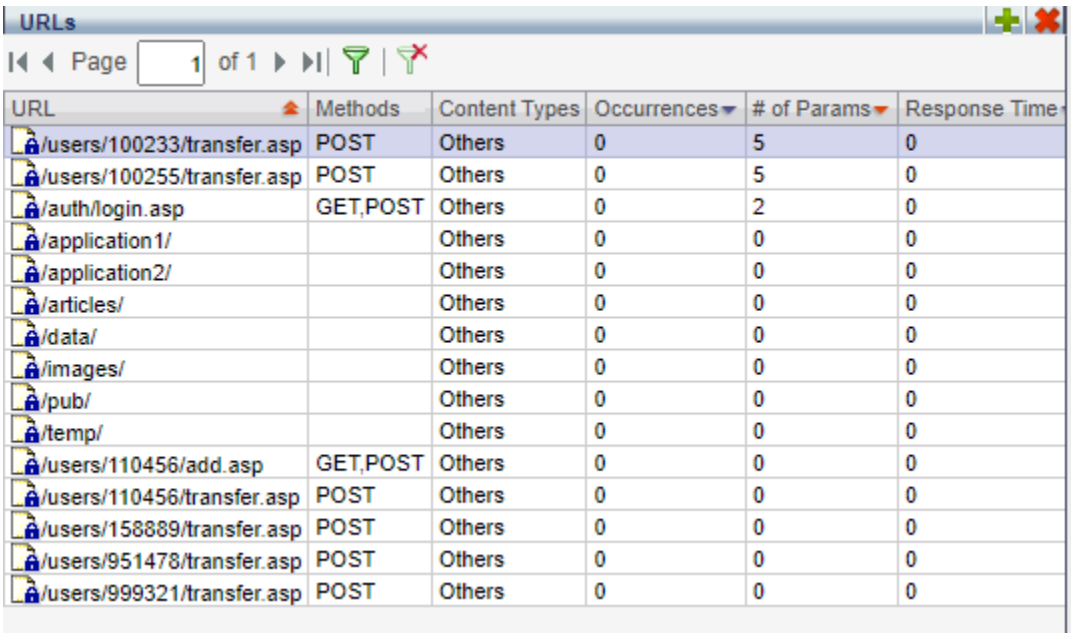


Figure 11

The URL list will collapse upon clicking Save (Figure 12) and move the URLs that match the pattern to the "URL Patterns" section. (Figure 13)



URL	Methods	Content Types	Occurrences	# of Params	Response Time
/users/100233/transfer.asp	POST	Others	0	5	0
/users/100255/transfer.asp	POST	Others	0	5	0
/auth/login.asp	GET,POST	Others	0	2	0
/application1/		Others	0	0	0
/application2/		Others	0	0	0
/articles/		Others	0	0	0
/data/		Others	0	0	0
/images/		Others	0	0	0
/pub/		Others	0	0	0
/temp/		Others	0	0	0
/users/110456/add.asp	GET,POST	Others	0	0	0
/users/110456/transfer.asp	POST	Others	0	0	0
/users/158889/transfer.asp	POST	Others	0	0	0
/users/951478/transfer.asp	POST	Others	0	0	0
/users/999321/transfer.asp	POST	Others	0	0	0

Figure 12

Click URL Patterns to review any and all patterns that have been created. (Fig 14)

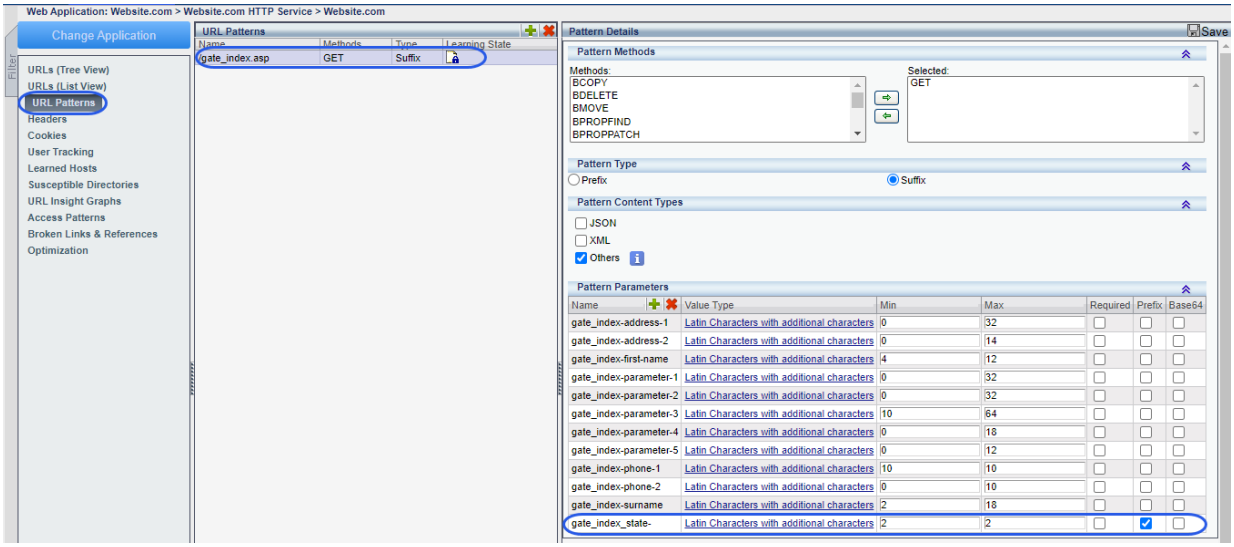


Figure 13

Now that we've tuned our Web Profile we can go back to the Optimization screen to confirm our issues have been resolved. Click **Optimization** and then click **Detect Optimization Issues**. (Figure 14) A message will be displayed indicating it may take a while to detect optimization issues. Click **OK** to proceed.

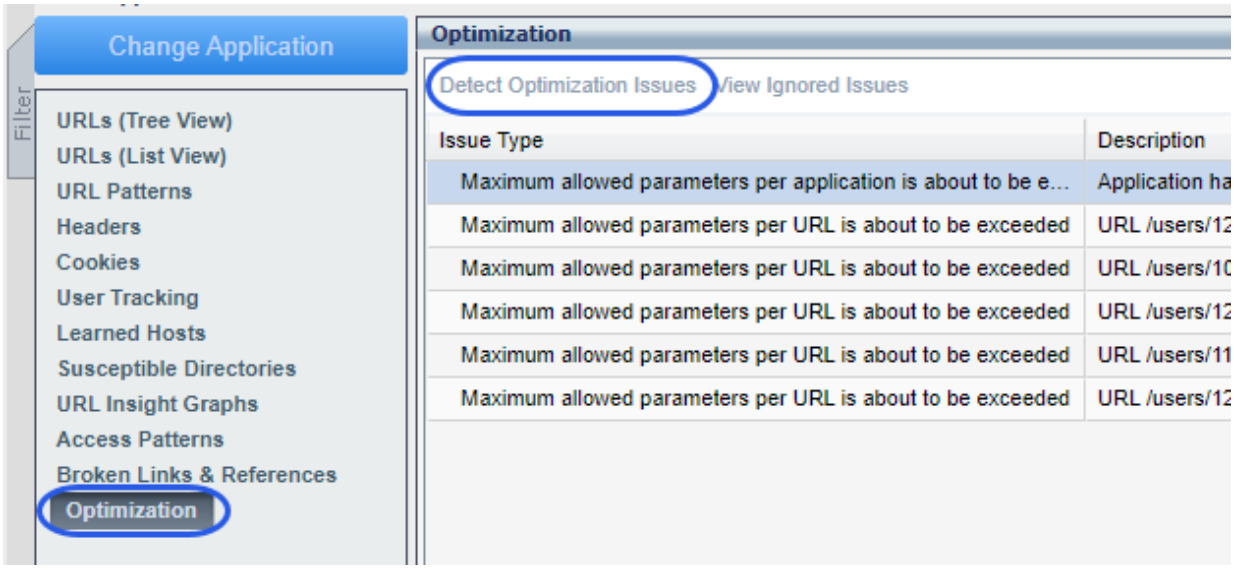


Figure 14

Once the job completes (usually happens in a matter of seconds) we can see there are no longer any pending issues.

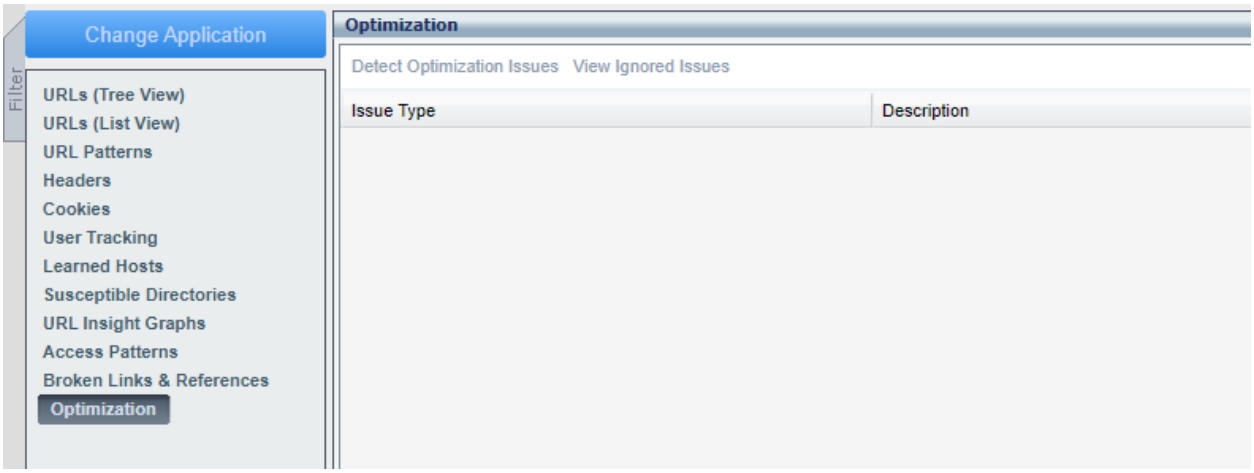


Figure 15

Congratulations on tuning your web profile!

I hope this article has equipped you with the knowledge to begin web profile tuning in your environment.