APPTRANA PROTECTION

RULES COVERAGE REPORT

Summary:

Most WAF solution fails, as application security is complex and creating rules in-house is a time-consuming job which requires expertise. Other Cloud security solutions that provides WAF generally go with cookie cutter solution. They provide certain generic rules and then provide customer means to write rules by themselves. It is up to the organizations to fine tune the rules to meet the application needs. Since default rules create false positives and fine-tuning rules becomes complex over time, organizations end up giving up on WAF compromising security for convenience.

We at Indusface approach the problem differently. We believe, security of the application starts with detection and AppTrana ensures that all the vulnerabilities are detected and we also ensure it is protected by expert written rules. Our experts fine-tune the rules based on the application need to avoid false positives and ensure that your application remain secure round the clock.

The following checklist gives you overview of rule coverage provided by AppTrana' s different rules.

Advance Rules: Rules which are fine tuned for FPs and are put in block mode from day zero.

Premium Rules: Rules which are applied to site and moved to block mode after monitoring traffic for 14 days ensuring there are no FPs

Custom Rules: Rules which are written for specific application needs in consultation with customer.

Note that we can have more variants of WAF rules in place for each category and only generic category and types are captured in this document.

Summary:

Category	Severity	Rule Type	Rule Description
Bot Protection	High	Advance	Good bot pretender detected.
DOS Protection	Critical	Advance	DoS/DDoS detected based on specified IP threshold value - 1.
DOS Protection	Critical	Advance	DoS/DDoS detected based on specified Cookie threshold value.
Cross-Site Scripting	Critical	Premium	Cross-Site Scripting attack attempt detected in HTTP request URI, Arguments and XML reuqests - 1.
Cross-Site Scripting	Critical	Advance	Cross-Site Scripting attack attempt detected in HTTP request Cookies and XML requests.
Cross-Site Scripting	Critical	Premium	Cross-Site Scripting attack attempt detected in HTTP request URI, Arguments and XML reuqests - 2.
HTTP Policy Violation	Medium	Premium	Non-supported HTTP request method (other than GET, POST & HEAD) detected.
HTTP Policy Violation	Low	Advance	Non-supported HTTP request headers detected.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI and arguments.
SQL Injection	Critical	Premium	SQL Injection attempt detected in HTTP request cookies and XML requests.
SQL Injection	Critical	Premium	SQL Injection attempt detected - 1.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 3.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 2.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request cookies or in XML requests.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments or HTTP Headers.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI and arguments.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request cookies and XML requests.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 1.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 2.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 3.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 4.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI, arguments or Cookie.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments or Cookie.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 5.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 4.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 6.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 7.

SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI, arguments or Request Headers.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments, HTTP Headers or XML file.
SQL Injection	Critical	Advance	Server side Injection attempt detected in HTTP request URI or arguments.
SQL Injection	Critical	Advance	Server side Injection attempt detected in HTTP headers or XML file.
Bot Protection	High	Premium	Automated program based User-Agent/HTTP header detected.
Bot Protection	High	Advance	Security scanner related User-Agent detected.
Bot Protection	High	Advance	Security scanner related HTTP header detected.
Bot Protection	High	Advance	Security scanner Nessus based URI detected.
Bot Protection	High	Advance	Website Crawler related User-Agent/HTTP header detected.
Bot Protection	High	Advance	Website Crawler related User-Agent/HTTP header (from internal database) detected.
Bot Protection	High	Advance	Security scanner related User-Agent/HTTP header (from internal database) detected - 2.
Bot Protection	High	Advance	Security scanner related User-Agent/HTTP header (from internal database) detected - 1.
Bot Protection	High	Advance	Security scanner related URI detected.
Bot Protection	High	Advance	Command Line Tool/Library related User-Agent/HTTP header (from internal database) detected.
Bot Protection	High	Advance	Bad bots related User-Agent/HTTP header (from internal database) detected.
File Injection	High	Advance	File injection attempt detected in HTTP request URI and arguments.
File Injection	High	Advance	File injection attempt detected in HTTP request header and XML requests.
OS Command Injection	Critical	Advance	System command injection attempt detected - 1.
OS Command Injection	Critical	Advance	System command injection attempt detected - 2.
PHP Injection	High	Advance	PHP injection attempt detected in HTTP request URI and arguments.
PHP Injection	High	Advance	PHP injection attempt detected in HTTP request header and XML requests.
Bot Protection	High	Advance	Bad reputated IP detected.
File Inclusion	High	Advance	Local File Inclusion (LFI) attempt detected via file traversal character sequences.
File Inclusion	High	Advance	Local File Inclusion (LFI) attempt detected via "\\\" character sequences.
File Inclusion	High	Premium	Local File Inclusion (LFI) attempt detected using path pointing from root directory.
Abuse of Functionality	Medium	Advance	Base64-encoded payload detected in HTTP request.
File Inclusion	Medium	Premium	Remote File Inclusion (RFI) attempt detected.
Abuse of Functionality	Medium	Advance	JavaScript encoding abuse detected - 1.
Abuse of Functionality	Medium	Advance	JavaScript encoding abuse detected - 2.

Recmote Code Execution	High	Advance	GNU Bash remote code execution (CVE-2014-6271) detected - 1.
Recmote Code Execution	High	Advance	GNU Bash remote code execution (CVE-2014-6271) detected - 2.
Recmote Code Execution	High	Advance	GNU Bash remote code execution (CVE-2014-6271) detected - 3.
Recmote Code Execution	High	Advance	GNU Bash remote code execution (CVE-2014-6271) detected - 4.
HTTP Response Splitting	Medium	Advance	HTTP response splitting attempt detected in HTTP request cookies - 1.
HTTP Response Splitting	Medium	Advance	HTTP response splitting attempt detected in HTTP request cookies - 2.
HTTProxy Protection	Medium	Advance	HTTP Proxy request header detected.
Recmote Code Execution	High	Advance	Remote code execution attempt via suspicious Java class detected. User can execute system commands via processbuilder or runtime calls and an attacker can misuse these classes submitting improperly sanitized objects to run malicious system commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2018-11776 and CVE-2017-5638) in Apache Struts via suspicious Java class detected. The vulnerability exists in the core of Apache Struts due to improper validation of user-provided untrusted inputs under certain configurations causing remote code execution.
Recmote Code Execution	Medium	Advance	Apache Tomcat Remote Code Execution (CVE-2019-0232) attack attempt detected.
DOS Protection	Critical	Advance	DoS attack using Ping headers in HTML5 - 1.
DOS Protection	Critical	Advance	DoS attack using Ping headers in HTML5 - 2.
DOS Protection	Critical	Advance	DoS attack using Ping headers in HTML5 - 3.
DOS Protection	High	Premium	DoS attack using Ping headers in HTML5 - 4.
Recmote Code Execution	High	Advance	Microsoft IIS HTTP.sys Remote Code Execution Exploit attempt (CVE-2014-6321)
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using echo and expr commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using variations of grep commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using variations of grep commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during

			file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using cc or wget commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using linux system commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
Recmote Code Execution	High	Advance	Remote code execution attempt (CVE-2017-5638) using windows system commands in Apache Struts via content-type request header detected. The Jakarta Multipart parser in Apache Struts has incorrect exception handling and error-message generation during file-upload attempts, which allows remote attackers to execute arbitrary commands.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI and arguments.
SQL Injection	Critical	Premium	SQL Injection attempt detected in HTTP request cookies and XML requests.
SQL Injection	Critical	Premium	SQL Injection attempt detected - 1.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 3.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 2.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request cookies or in XML requests.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments or HTTP Headers.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI and arguments.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request cookies and XML requests.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 1.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 2.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 3.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected - 4.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI, arguments or Cookie.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments or Cookie.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 5.

SQL Injection	Critical	Advance	SQL Injection attempt detected - 4.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 6.
SQL Injection	Critical	Advance	SQL Injection attempt detected - 7.
SQL Injection	Critical	Advance	Blind SQL Injection attempt detected in HTTP request URI, arguments or Request Headers.
SQL Injection	Critical	Advance	SQL Injection attempt detected in HTTP request URI, arguments, HTTP Headers or XML file.
SQL Injection	Critical	Advance	Server side Injection attempt detected in HTTP request URI or arguments.
SQL Injection	Critical	Advance	Server side Injection attempt detected in HTTP headers or XML file.

Apart from this, specific custom rules can be written to address application specific needs. These rules are again created by Indusface security experts. Certain use cases that can be addressed are provided below, please note these are not comprehensive and should be used to judge the type of use cases that can be addressed through AppTrana.

Theft/DLP Protection:

Customers who need to protect sensitive information protected and ensure certain information do not leave the organization can request for response based rule, which would monitor their response traffic and mask sensitive data. When these rules are enabled, sensitive information will be masked on the logs as well.

Note: Response based rules are highly intrusive and should be enabled judiciously as it may affect functioning of the application.

BAD IP Protection:

Indusface provide IP protection which will show IP's which are malicious, customers can choose to either to monitor these malicious IP's manually or have automated rule enabled which would block these IP's automatically. IP's with bad reputation is identified by using internal Global Threat Platform which identifies malicious IP's based on behaviour across all sites under Indusface Protection.
Apart from this Global Threat Platform also gets periodic updates from Global 3rd party database which marks certain IP malicious.

Customer can also choose to have TOR IP's blocked through custom rule.

Protection Against Hidden Form Fields:

Incase customers have any hidden form fields and want to restrict requests which sends out of bound values for the field , then customer can request for custom rule which would be written by our security experts based on their need.

File Upload Violation:

Customers based on application need can request for custom rule written to avoid file uploads that does not meet the acceptable parameters.

Positive Security Rules:

Customer can choose to enable positive security model, in which some or all negative model rules would be disabled for the customer based on their need and positive security rules created which would take into accepted values for various fields like URLs, directories, cookies, headers, form/query parameters, File upload Extensions, Allowed metacharacters etc and allow only values that meets the accepted parameters.

HoneyPot Bot Defender Rule:

We have enhanced our Bot defender rules which can now identify malicious bots through honeypots and block them. If a new malicious bot is identified when it attacks one of the protected site, this information will be registered in our global threat intelligence database and attack from same botnet on any other sites under our protection will be blocked faster.

Behaviour Rules:

We have sophisticated anomaly scoring/ behaviour rules that changes the protection status of rules based on certain behaviour observed in the application. This can be done at application level or at a specific page level.

Tampering Protection Policies:
Customers can also enable tampering policies which would help them against cookie tampering/poisoning attacks. It also protects application from tampering like URL rewriting, encryption tampering etc.. This rule can also be configured to protect against attacks to identify predictable resource location, unauthorized access, server reconnaissance.

HTTP Parameter Controlling Policies:
Solution protects HTTP Parameter pollution and tampering attacks and policies can be written to protect against HTTP parameter pollution attack, restricting/controlling HTTP methods and validating header length, content length, Body length, Parameter length, body line length etc.

Enterprise Features;

AppTrana supports all enterprise use cases including-

Support for Transformation Functions:
As part of core rules AppTrana supports transformation functions like URL Decoding, Null Byte string termination

Customized Error Message
Based on application requirement customer can request for rules to mask their server errors and show custom pages instead of default server errors.

Support for Custom Ports & Protocols:
By default the rules are written for HTTP/HTTPS traffic and WAF listens on port 80/443. Customers can request for additional custom ports be opened based on their need and monitoring of additional protocols like SOAP, XML etc.

Support for IPv6
Customer can enable IPv6 support for their sites by requesting it while onboarding. With this clients connecting to the application will be able to connect using IPv6 even if backend does not support IPv6.

Support for SIEM
SIEM API's are available that will enable customers to get real time attack logs from AppTrana that can be integrated with their SIEM tools for further analysis

Support for 2FA & RBCA
AppTrana provides support for Role based access control as well ensures access to AppTrana portal through 2FA support.

Note: Please reach out for any additional needs.