ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ТЕНДЕРУ № _____ «Закупка программного обеспечения «Сканер защищенности веб-приложений SolidWall DAST (Dynamic Application Security Testing)»

№ п/п	Наименование закупаемой позиции	Кол-во
1	Лицензия «SolidPoint DAST», SLA1-SPD (8x5).	18 целей сканирования

Обязательные условия (является 100% выполнение условий ТЗ)

1. Стоимость (фиксируется в рублях на день предоставления КП)
2. Версия 1.0. (или выше)
3. Срок действия лицензии 12 месяцев
4. В течении срока действия лицензии включено предоставление обновлений ПО

Т Е Х Н И Ч Е С К А Я С П Е Ц И Ф И К А Ц И Я

«Сканер защищенности веб-приложений SolidWall DAST (Dynamic Application Security Testing)»

1. ТРЕБОВАНИЯ К DAST

1.1. Нефункциональные требования:

- Система должна состоять из следующих компонент:
  - подсистемы сканирования, состоящей из узлов анализа, выполняющих непосредственные вызовы к тестируемым на безопасность приложений;
  - подсистемы управления и представления результатов анализа, обеспечивающей запуск, управление сессиями сканирования на узлах анализа и дальнейший просмотр результатов, и выгрузку отчетов;
  - подсистемы хранения данных, которая осуществляет централизованный сбор, и хранение данных подсистемы управления и представления результатов анализа и подсистемы сканирования;
- Система должна обеспечивать возможность установки каждого из компонентов системы на отдельных виртуальных и физических сетевых узлах для создания распределенных масштабируемых и отказоустойчивых конфигураций.
- Система должна поддерживать механизмы горизонтального масштабирования для распределения нагрузки.
- При необходимости масштабирования Система должна поддерживать добавление новых подсистем (узлов) без внесения изменений в код или архитектуру самой Системы и без прерывания в обслуживании.
- Все подсистемы должны работать в режиме отсутствия доступа к Интернет с возможностью периодического обновления.
- Система должна обеспечивать возможность поставки в виде дистрибутива устанавливаемого и настраиваемого локально у Заказчика.
- Система должна поддерживать круглосуточный режим функционирования. Допускается временная недоступность сервисов при проведении восстановительных (в случае сбоя) работ.
- При истечении срока действия технической поддержки Система не должна блокировать работу.
- В системе должны быть реализованы механизмы управления учетными записями пользователей с возможностью их модификации, блокирования и удаления из Системы.
- Система должна предусматривать блокирование встроенных и локальных учетных записей при настройке LDAP интеграции.
- В Системе должны быть реализованы функции протоколирования событий безопасности.
- Функция аудита должна однозначно сопоставлять каждое подлежащее аудиту событие с идентификатором пользователя, который был инициатором этого события с фиксацией времени события и определением, по возможности, адреса (например, сетевого).
- В рамках Системы должна быть реализована возможность, централизованного сбора и хранения журналов регистрации событий, либо поддерживаться интерфейс с решениями по сбору и хранению журналов, используемых в Компании.

1.2. 1.2 Функциональные требования:

- Система предназначена для динамического анализа безопасности как разрабатываемых, так и находящихся в эксплуатации веб-приложений и веб-сервисов Заказчика с целью выявления и устранения уязвимостей в интеграции с циклом разработки ПО Заказчика.
- Данная цель должна выполняться за счет наличия у внедряемой Системы следующих ключевых возможностей:
  - Программное обеспечение должно быть включено в Единый реестр российских программ для электронных вычислительных машин и баз данных.
  - Система должна обеспечивать возможность построения отчетности по результатам анализа в формате pdf.
  - Система должна обеспечивать многопользовательский ролевой доступ.
  - Система должна иметь встроенные средства идентификации и аутентификации.
  - Система должна иметь возможность интеграции с LDAP/AD.
  - Система должна поддерживать разделение зон доступа (мультитенантность).
  - Система должна поддерживать возможность указания LDAP сервиса для каждого тенанта

- В рамках тенанта должна обеспечиваться возможность указания ограничения сканируемых ресурсов
- Обеспечение журналирования выполняемых операций (ведение аудит-логов).
- Система должна иметь возможность встраивания в непрерывный процесс разработки ПО (CI/CD).
- Система должна иметь возможность сканировать веб приложения, в том числе SPA.
- Система должна иметь возможность сканировать сервисы на базе REST API.
- Система должна поддерживать создание шаблона сканирования с такими атрибутами как: адрес сканируемого приложения, методы аутентификации с данными, файл клиентского сертификата, файл OpenAPI спецификации, ограничения по скорости сканирования.
- Система должна позволять сканировать приложение с указанием данных непосредственно, с автоматической генерацией шаблона.
- Система должна позволять сканировать приложение с указанием идентификатора соответствующего шаблона.
- Система должна позволять выбирать типы сканирования, и типы проводимого анализа при запуске сканирования
- Система должна поддерживать возможность одновременного запуска нескольких параллельных процессов сканирования, в том числе в распределенной архитектуре.
- Система должна поддерживать черный список URL при сканировании приложения с помощью регулярных выражений.
- Система должна поддерживать возможность сканировать приложения с прохождением аутентификации.
- Система должна обеспечивать поддержку технологий обновления короткоживущих сессий.
- Система должна поддерживать следующие методы обнаружения потенциальных точек ввода данных при сканировании: анализ разметки и переход по ссылкам, перебор URL по словарю, статико-динамический анализ клиентского JS кода.
- Система должна поддерживать возможность получения информации о точках ввода данных с помощью импорта спецификации Open API.
- Система должна обеспечивать выявление компонентов стека прикладного ПО анализируемого веб-приложения на основе сигнатур.
- Система должна обеспечивать выявление как минимум следующих классов угроз:
  - Уязвимости стандартных компонентов приложений и инфраструктуры их доставки;
  - Weak Credentials (Использование слабых паролей);
  - SQL Injection (Внедрение команд языка SQL-запросов);
  - NoSQL Injection (Внедрение команд языка нереляционных БД);
  - XSS (Межсайтовая подделка запросов);
  - XXE (Включение внешних сущностей XML);
  - Insecure Deserialization (Небезопасная десериализация);
  - JS Prototype Pollution (Изменение прототипа базовых объектов);
  - JS DOM Based XSS (XSS с использованием DOM браузера);
  - Path Traversal (Доступ к файлам и директориям через обход пути);
- Система должна содержать детальное описание найденных уязвимостей.
- Система должна проводить инвентаризацию загружаемых JS скриптов, включая сторонние, с представлением в иерархическом формате.
- Система должна сопоставлять загружаемые JS скрипты с подключаемыми через них файлами cookie.
- Система должна иметь возможность осуществлять экспорт табличных данных в машиночитаемых форматах (JSON или CSV).
- Система должна поддерживать возможность подключения дополнительных модулей анализа уязвимостей.
- Система должна обеспечивать возможность классификации выявленных уязвимостей по типу и степени критичности.
- Система должна иметь возможность установку в закрытом контуре (on premise)
- Система должна проводить проверку как внутренних адресов, так и публичных.
- Система должна поддерживать гипервизоры: Open stack, Vmware. Размещение в облачных сервисах.
- Система должна поддерживать управление через запросы API.