METASCAN

Сканер уязвимостей для вашей инфраструктуры

Защищаем от потери денег и критически важных данных в результате кибератак

Ежедневно проводим автоматизированное сканирование всех ваших IP и доменов.

Анализируем отчёты и проводим дополнительное ручное тестирование уязвимостей.

Еженедельно проводим планёрки с командой ИБ и выстраиваем процесс управления уязвимостями ИБ внешнего сетевого периметра.

ВОЗМОЖНОСТИ СКАНЕРА

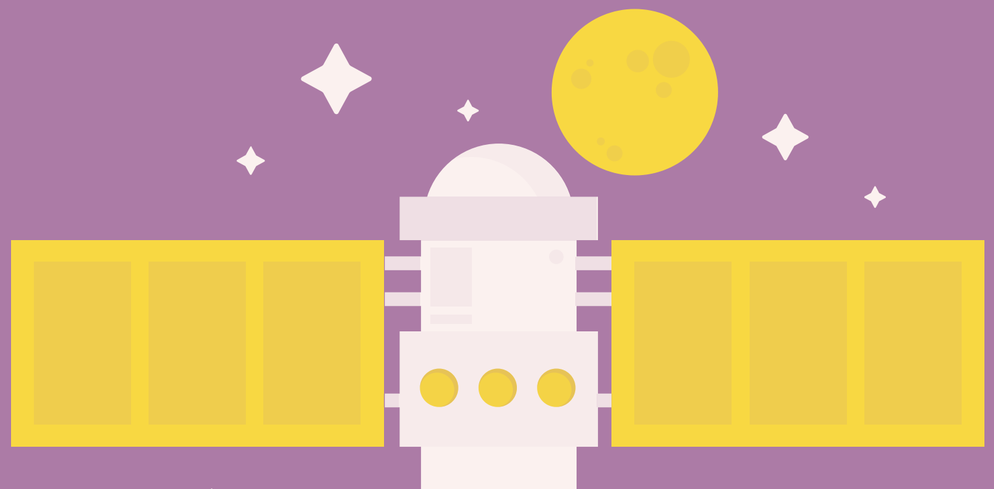
- Контроль портов периметра на соответствие whitelist
- Проверка ППО по базе в 110 807 уязвимостей и эксплойтов. MITRE, NIST + ExploitDB
- Проверка на OWASP-TOP-10 веб-уязвимости (XSS, SQLi, т.д)
- Поддержка возможностей HTML5, AJAX, Angular, React
- Подбор слабых паролей по 40 протоколам
- Сканирование веб-порталов с аутентификацией
- Поиск уязвимостей в CMS и версиях JS-библиотек
- Тестирование API на основе swagger-описания
- Обнаружение административных интерфейсов (PHPMyadmin, PGAdmin, Kubernetes, и д.р.)

L3-L7 проверки

БЕРЕЖЕТ ВАШЕ ВРЕМЯ

ПРЕИМУЩЕСТВА

- Автоматическое обнаружение доменов и сетей компании
- Графическое отображение угроз и "светофор" для руководства
- Интеграция с SIEM или IRP через SYSLOG или REST API
- Возможность ежедневно проверять до 50 000 целей
- Выгрузка данных в форматах JSON, CSV
- Интеграция ваших модулей и сканеров специфичных для вашего ПО
- Отправка уязвимостей в систему из ваших источников



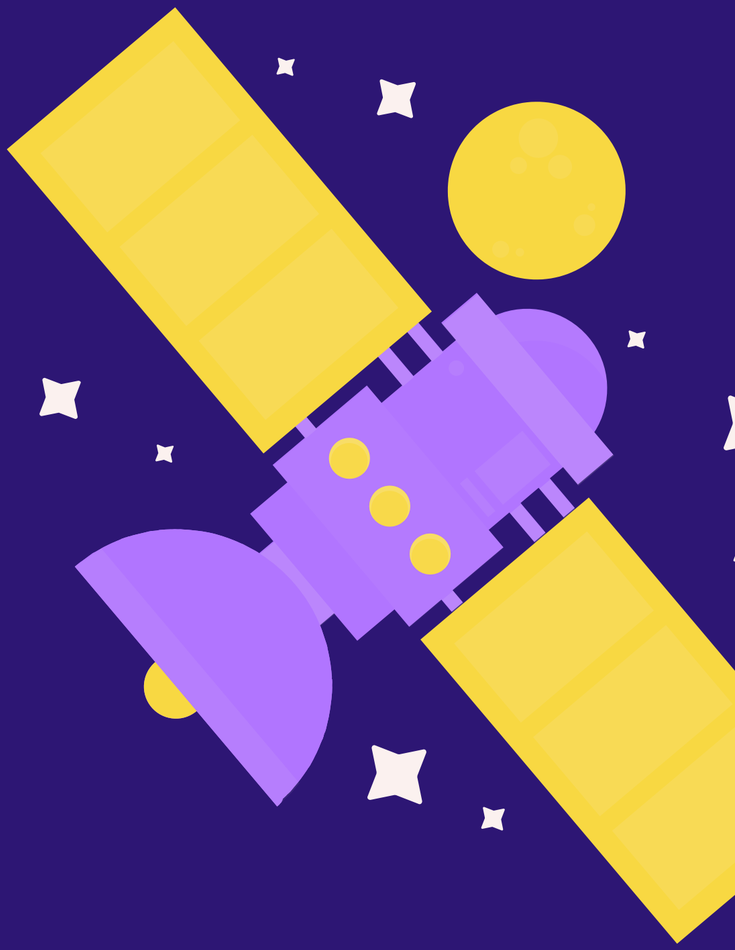
СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

Для Enterprise

- Ввод систем в эксплуатацию
- Проверка на уязвимости как часть релизного цикла ПО
- Проверка после изменений сетевой инфраструктуры
- Проверки на соответствие регламентам
- Защита от эпидемий, 1-day эксплойтов (Log4j, BadRabbit, ProxyLogon)

ИНТЕГРАЦИЯ С SIEM

- Источник информации о появлении новых доменов.
- Источник информации о открывающихся портах.
- Источник информации о L3-L7 уязвимостях
- Передача событий по syslog.
- Возможность постановки задач через HTTP-API.



Python wrapper

ДОБАВЛЯЙТЕ СВОИ СКАНЕРЫ И СКРИПТЫ

```
class Scanner(object):
    name = "scanner_base"
    vuln_type = "default_vuln_type"
    user_options = {}
    Vulnerability_body_fields_to_web_interface = []

    def __init__(self, opts, target, metadata):
        self.metadata = metadata
        self.opts = opts
        self.target = target

    @staticmetod
    def circuit(Metadata):
        '''
        Логика работы сканера.
        Принимает на вход объекты типа Metadata.
        Результатом работы должны быть экземпляры класса CVE.
        '''
        return [Vulnerability(), Vulnerability()]

    def check_start_condition(self):
        '''
        Проверка параметров, которым должен соответствовать Target для запуска сканера
        True, если сканер должен запуститься. В другом случае False.
        '''
        return True

    class ScannerError(Exception):
        def __init__(self, value):
            self.value = value
        def __str__(self):
            return repr(self.value)
```

METASCAN.RU

DO@METASCAN.RU
+7 916 986 14 00
+7 495 152 13 37