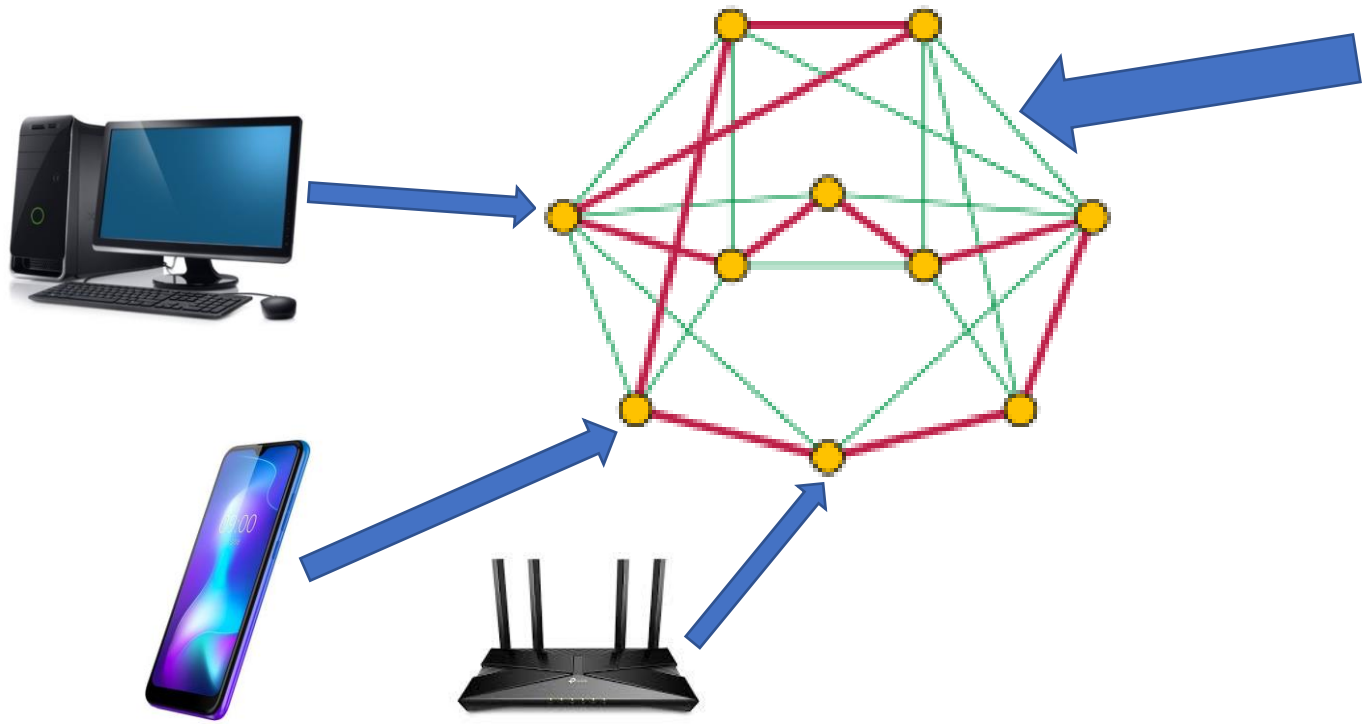
Как детектируется вредоносное программное обеспечение

Андрей Калегин, Research Software Engineer, Kaspersky

Определения

- ИБ - практика предотвращения несанкционированного доступа, использования, раскрытия, искажения, изменения, исследования, записи или уничтожения информации
- Основная задача информационной безопасности — сбалансированная защита конфиденциальности, целостности и доступности данных

С чем работаем



Модель OSI

Данные	Уровень	
Данные	Прикладной	доступ к сетевым службам
Данные	Представления	представление и кодирование данных
Данные	Сеансовый	Управление сеансом связи
Блоки	Транспортный	безопасное и надежное соединение точка-точка
Пакеты	Сетевой	Определение пути и IP (логическая адресация)
Кадры	Канальный	MAC и LLC (Физическая адресация)
Биты	Физический	кабель, сигналы, бинарная передача

Что делает ПО для ИБ

- Собирает и накапливает первичную информацию
- Анализирует (detect), накапливает результаты анализа
- Реагирует (response): удаляет, лечит, отправляет в карантин, разрывает соединение, изолирует узел и т.д.
- Опционально: передаёт свои результаты на анализ живым людям на вершине пирамиды принятия решений (SOC)
- Опционально: интегрируется с внешними системами

Что интересует на узлах

- Файлы
- Записи в БД (реестр Windows, например)
- Автозапуск
- Сервисы
- Сертификаты
- Содержимое ROM BIOS
- Периодически запасающиеся задачи (task manager, cron и т.д.)
- …..

- Действия запущенных программ

Домены

- Код
- Строки
- Ресурсы
- Экспорты/импорты
- Цифровая подпись
- Содержимое памяти работающего процесса
- …

Грубо говоря, домены – это совокупность текстов, в которых мы ищем интересующие нас паттерны

Yara

```
rule silent_banker : banker
{
    meta:
        description = "This is just an example"
        threat_level = 3
        in_the_wild = true
    strings:
        $a = {6A 40 68 00 30 00 00 6A 14 8D 91}
        $b = {8D 4D B0 2B C1 83 C0 27 99 6A 4E 59 F7 F9}
        $c = "UVODFRYSIHLNWPEJXQZAKCBGMT"
    condition:
        $a or $b or $c
}
```

Katran

```
int    falseAlarm;
$sig  delphi("'RegOpenKeyEx(0x80000001,"+
                "\"Software\\\\Borland\\\\Delphi\\\\Locales\",,,);'",GEN);

$on (delphi)
{
      falseAlarm = 1;
}

$sig evil    ("'LoadLibrary**(\"'*[10]i'?.exe'",GEN);

$on (EMULATOR_END_PROCESS_OBJECT && evil && falseAlarm == 0)
{
      SUSPICION_Trojan_Generic;
}
```

Suricata

drop tcp $HOME_NET any -> $EXTERNAL_NET any (msg:”ET TROJAN Likely Bot Nick in IRC (USA +..)”; flow:established,to_server; flowbits:isset,is_proto_irc; content:”NICK “; pcre:”/NICK .*USA.*[0-9]{3,}/i”; reference:url,doc.emergingthreats.net/2008124; classtype:trojan-activity; sid:2008124; rev:2;)

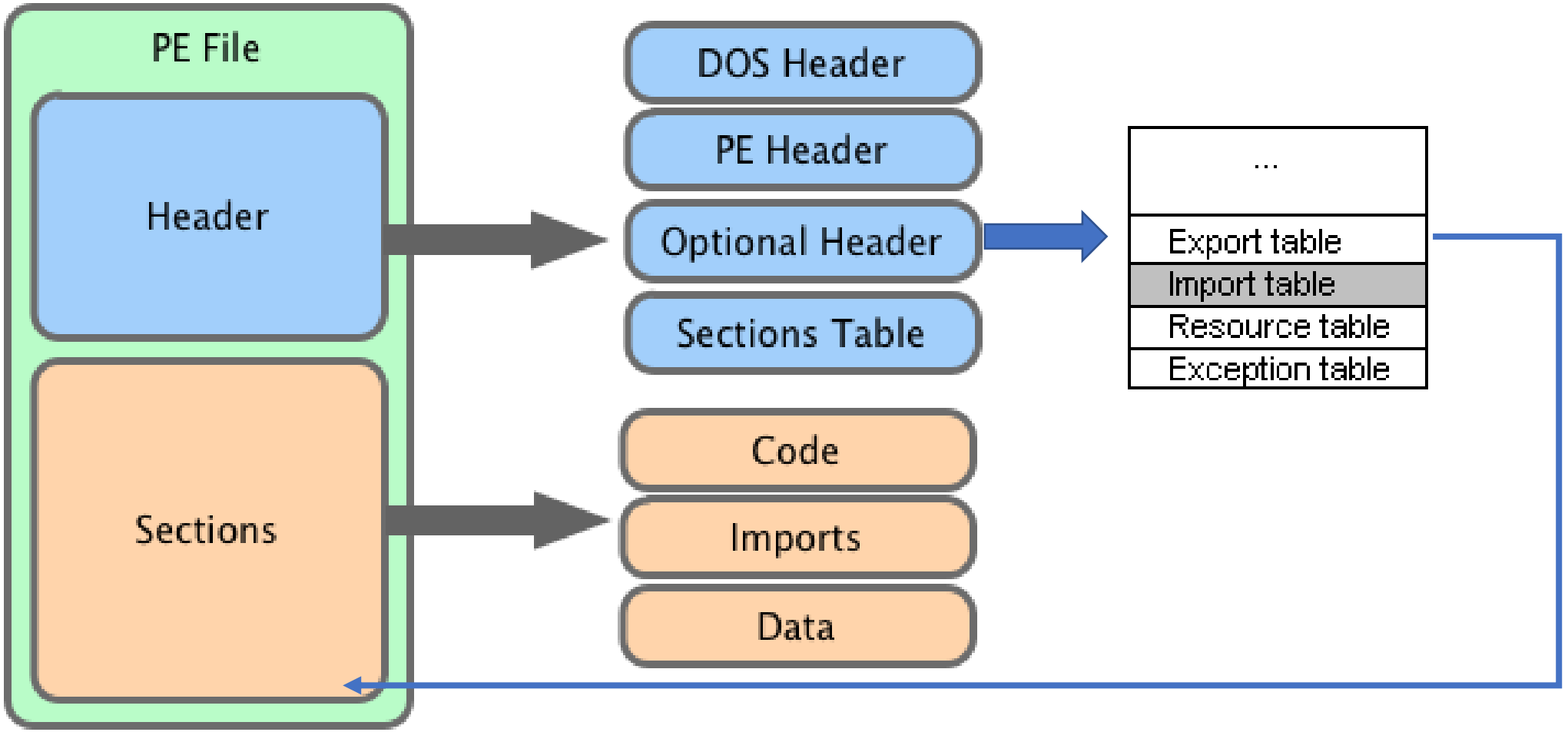
(Антивирусные) базы

- **Движок** наполняет домены
- К содержимому доменов применяются **правила**
- **А так же произвольный код, которому доступны внутренние api движка**
- Детектируем, выносим вердикт, реагируем
- Профит

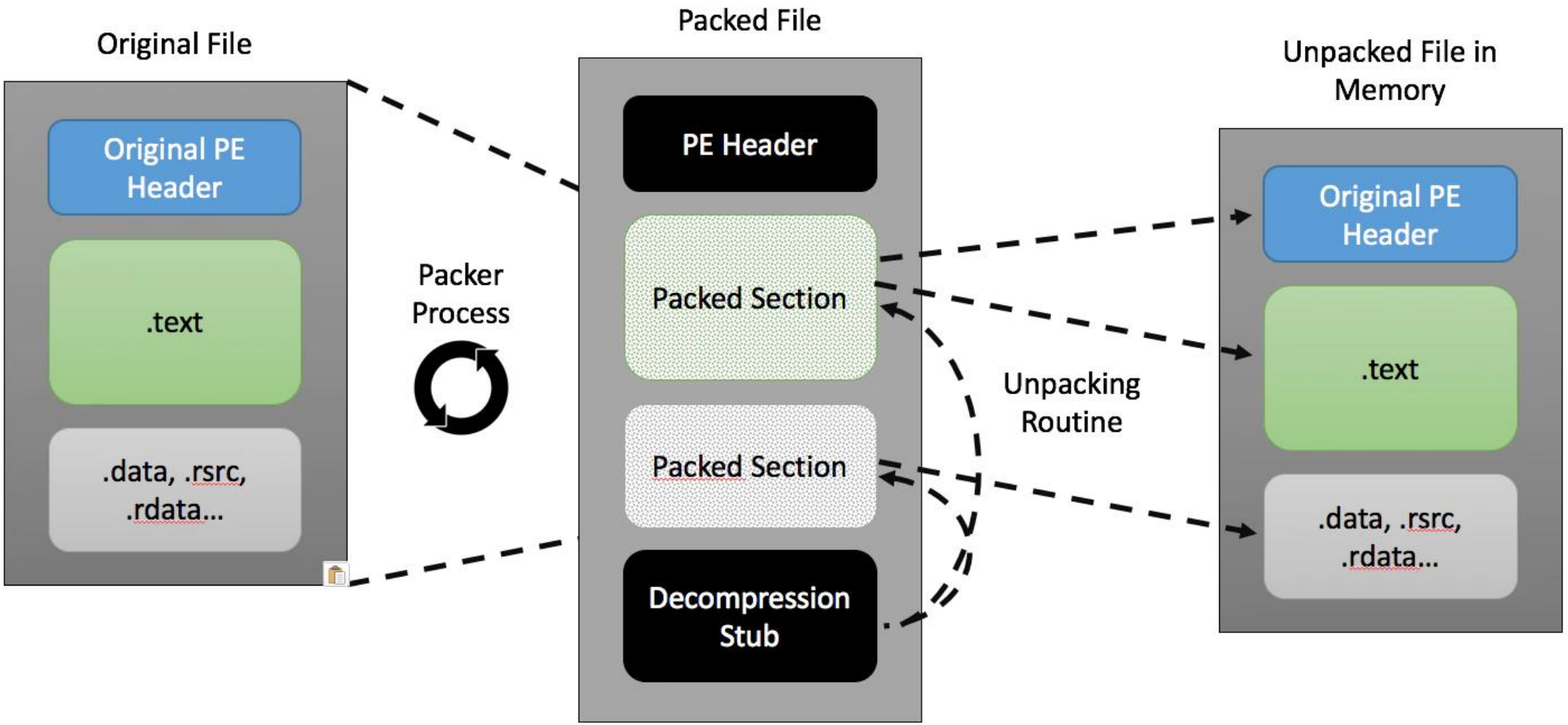
Статический анализ (исполняемого) файла

- Определение того, с чем мы вообще имеем дело
- Опционально: распаковка (архивы, в т.ч. с паролями, упакованные PE файлы)
- Извлечение признаков, сигнатурный анализ
- Извлечение признаков, анализ ML моделями
- Locality sensitive hashing
- Reverse engineering

Структура PE файла



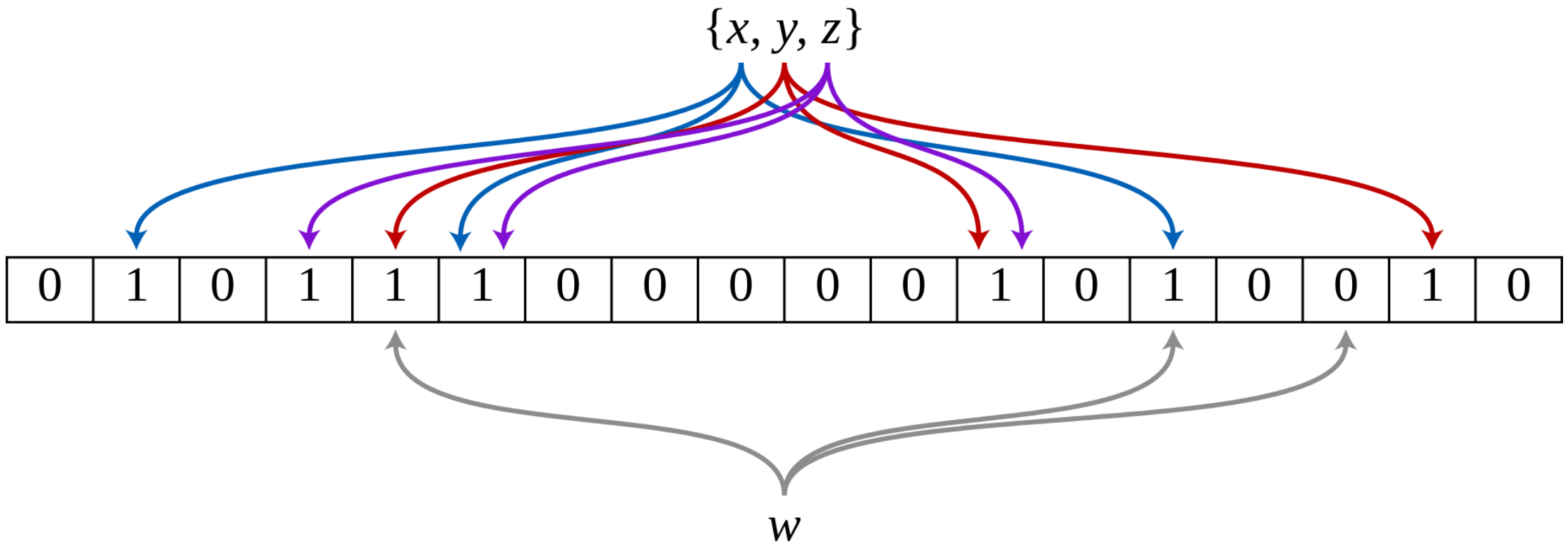
Упаковщики


Original File
Original PE Header
.text
.data, .rsrc, .rdata...
Packer Process
Packed File
PE Header
Packed Section
Packed Section
Decompression Stub
Unpacking Routine
Unpacked File in Memory
Original PE Header
.text
.data, .rsrc, .rdata...

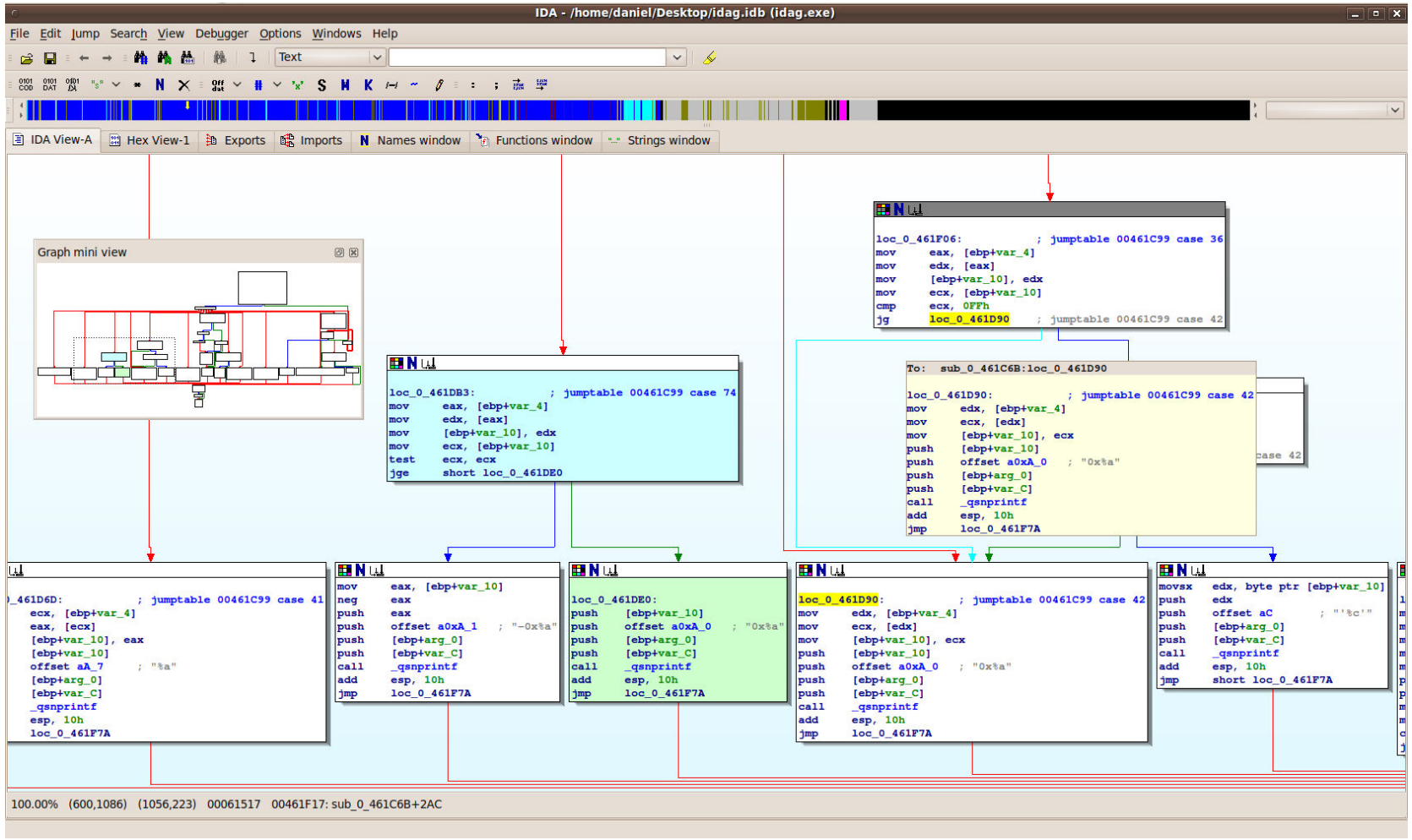
Извлечение признаков

- Извлечение содержимого хедеров, в т.ч. data directory
- Побайтовая статистика секций
- Извлечение строк из секций
- Разбор ресурсов конкретных форматов (Visual Studio, Delphi, Qt, Go)
- Нахождение участков исполняемого кода и извлечение признаков из него
- …
- Возможно, кладём в bloom-фильтр

Bloom filter



Reverse engineering



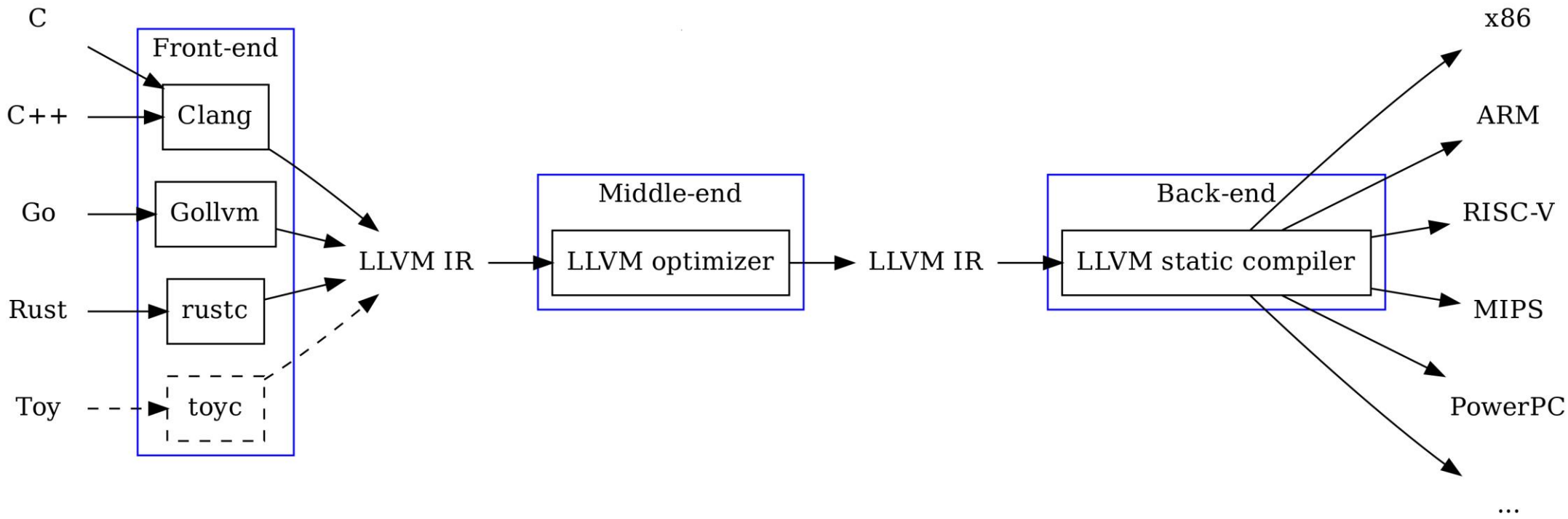
Бинарная обфускация

Obfuscated	Simplified
`a & (a^n)`	`a & ~n`
`a^b^c^a^d^c`	`b^d`
`(a-1) * a & 1`	`0`
`(x & n) | (~x & ~n)`	`x ^ ~n`
`(a && b) || (a ^ b)`	`(a || b)`
`(a && b) || (!a ^ !b)`	`(a || b)`

Бинарная обфускация

```
int main()
{
    int a = 1;
    while (a != 987)
    {
        switch (a)
        {
        case 1:
            a = 657;
            break;
        case 89:
            std::cout << "Step 2\n";
            a = 14;
            break;
        case 14:
            a = 987;
            break;
        case 39:
            a = 89;
            break;
        case 657:
            a = 33;
            break;
        case 33:
            std::cout << "Step 1\n";
            a = 39;
        }
    }
}
```

IDA Pro Microcode API: a-la LLVM IR в другую сторону



Динамический анализ

- Перехваты API вызовов (поведенческое детектирование)
- Сканирование памяти работающего процесса
- Построение графов запуска
- Анализ сетевых запросов
- Event log, AMSI и т.д.
- Работа в отладчике

- На живой системе
- Под эмулятором
- В sandbox’е

Перехваты

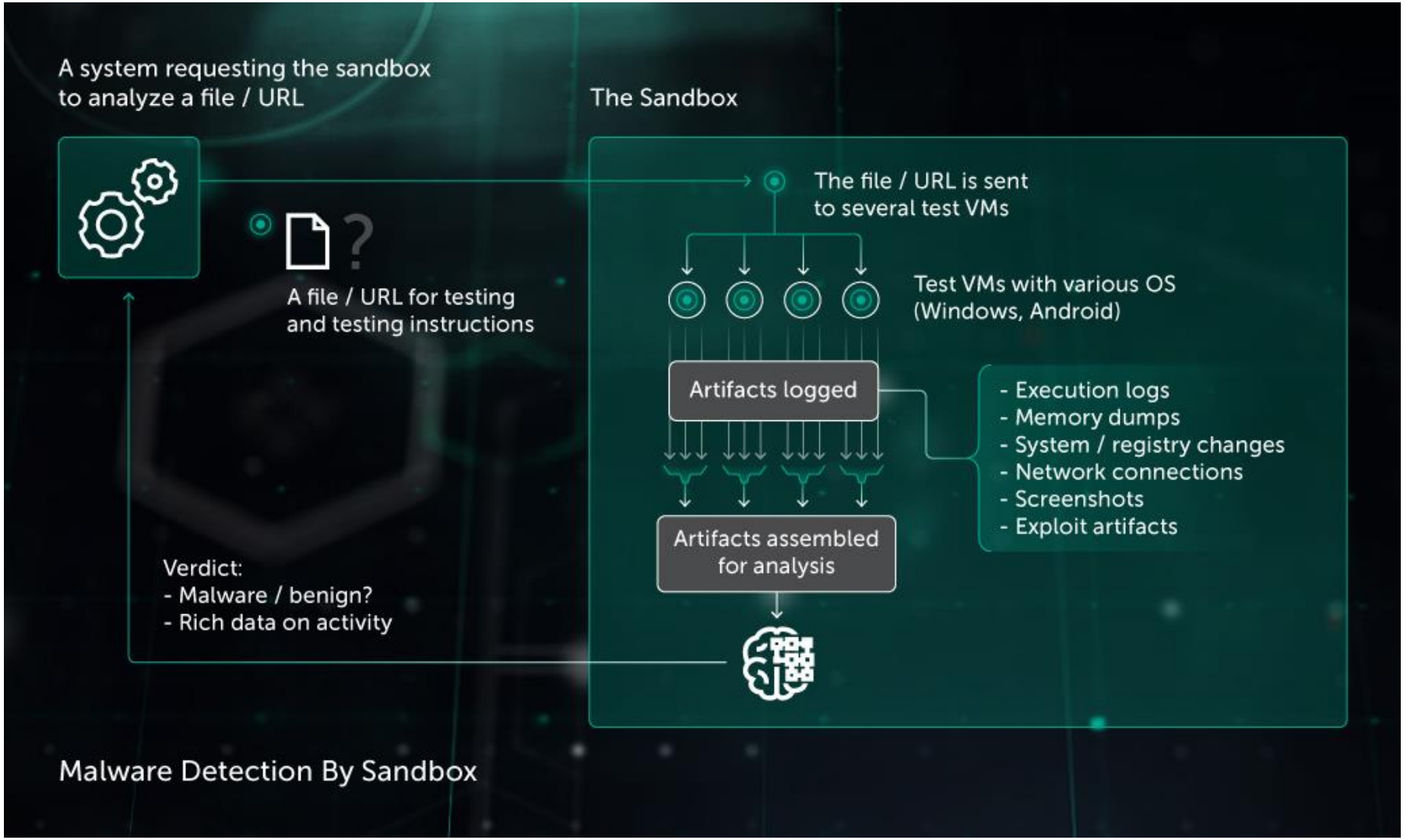
- Хуки
- Драйвера
- API (event log, amsi)

Эмуляция

- Бинарный эмулятор
- Скриптовые эмуляторы

- Inlab и на стороне пользователя

Sandbox

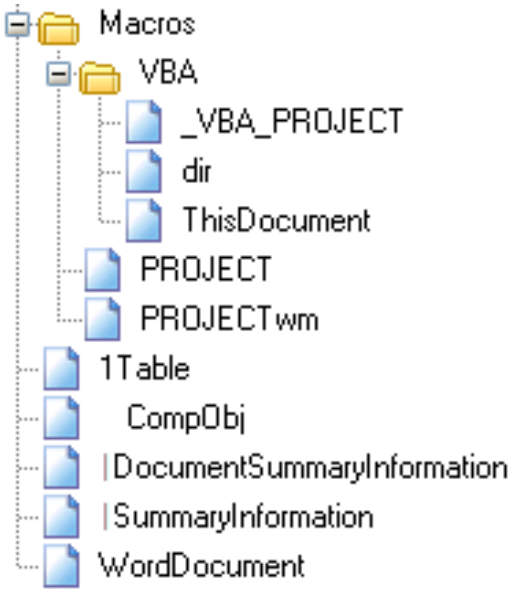

A system requesting the sandbox to analyze a file / URL
The Sandbox
The file / URL is sent to several test VMs
A file / URL for testing and testing instructions
Test VMs with various OS (Windows, Android)
Artifacts logged
- Execution logs
- Memory dumps
- System / registry changes
- Network connections
- Screenshots
- Exploit artifacts
Artifacts assembled for analysis
Verdict:
- Malware / benign?
- Rich data on activity
Malware Detection By Sandbox

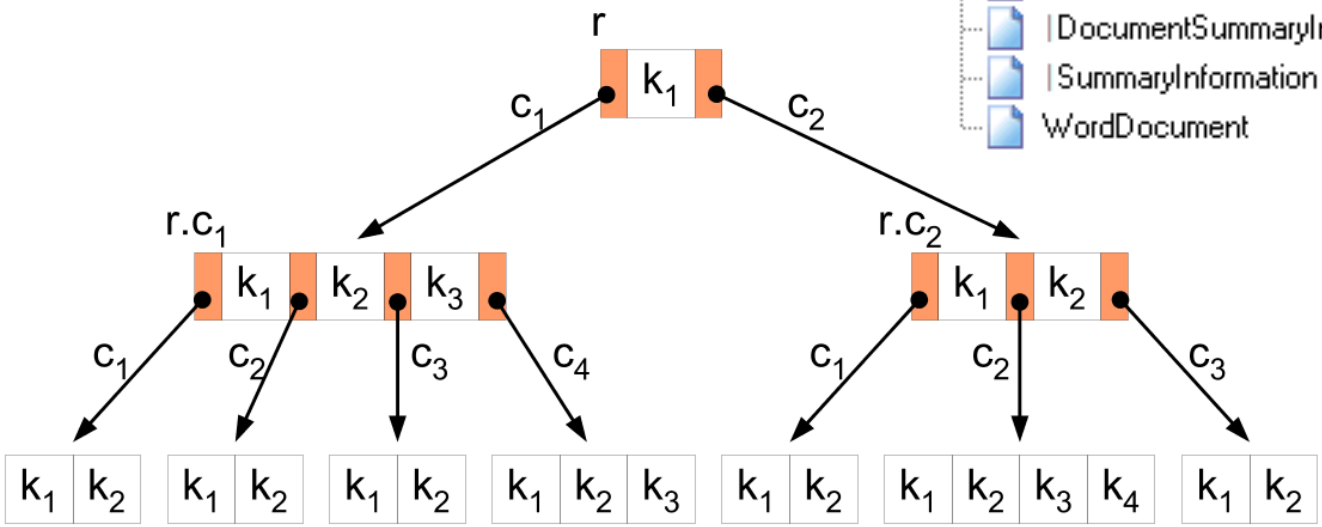
ВПО на скриптовых языках

- Извлечение: разбор форматов офисных документов
- Деобфускация
- Парсинг и извлечение структурных признаков
- Методы NLP: программа – это в первую очередь текст. Частоты символов, TF/IDF, подсчёт ngram.

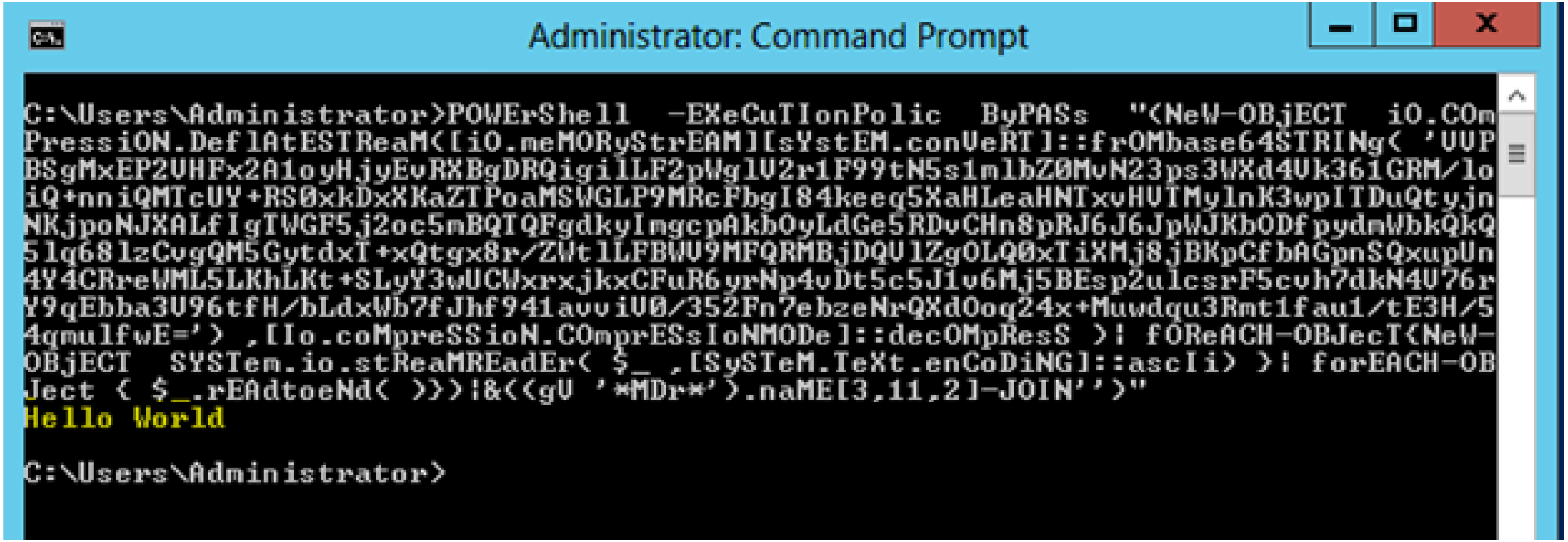
Извлечение скриптов

Type: /Catalog
Referencing: 7 0 R, 11 0 R
<<
/Type /Catalog
/Pages 7 0 R
/Names 11 0 R
/OpenAction
<<
/S /JavaScript
/JS '(function11\\(\\)'
;)
>>
>>

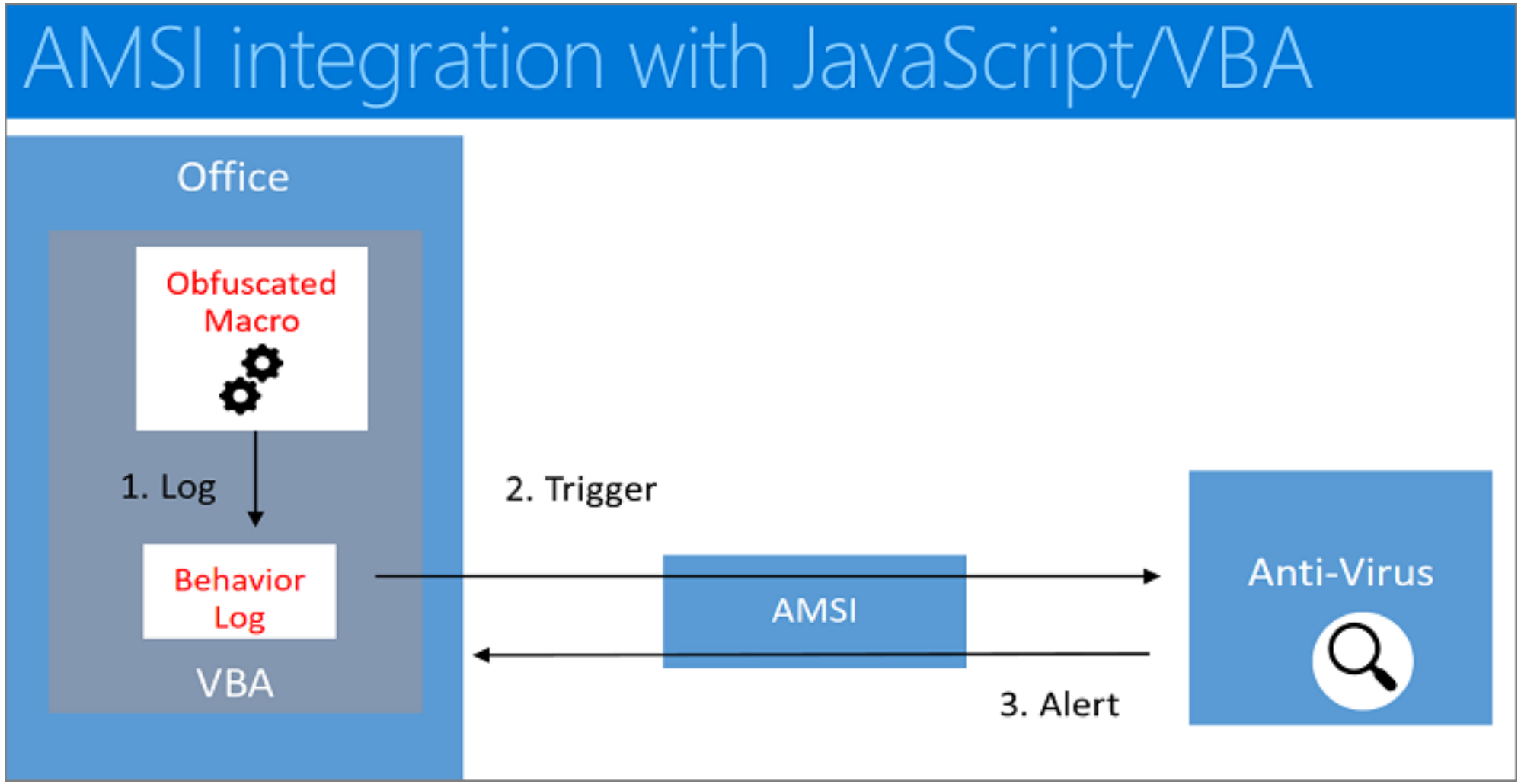




Обфускация скриптов



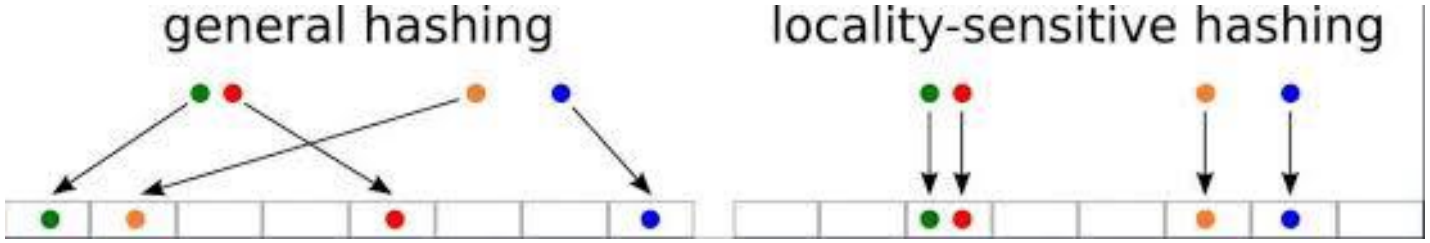
AMSI


AMSI integration with JavaScript/VBA
Office
Obfuscated Macro
1. Log
Behavior Log
VBA
2. Trigger
AMSI
3. Alert
Anti-Virus

ML

- Обучаем модели на основе +- тех же признаков, на которые смотрят люди.
- Аналитики и технологии автодетекта выступают своего рода толокой. Поток входящих обеспечивают silent’ы.
- Не столько детектируем новое, сколько ищем похожее на уже задетектированное технологиями, доступными только inlab
- White(allow)listing важен
- Ограничения реализации (многообразие платформ, отсутствие операция с плавающей точкой)
- Требования производительности

Locality sensitive hashing



- Вероятностный метод понижения размерности в многомерных данных
- В том числе сведение нечёткого поиска к запросам к key-value DB
- Cosin/MinHash/MinWise/…
- http://www.mmds.org/

Где применяется ML в ЛК

- Детектирование вредоносных исполняемых файлов и скриптов
- Детектирование вредоносных урлов и хостов на основе данных whois, pdns
- Детектирование спама и фишинга
- Для детектирования apk
- Для выделения подозрительных apk на основе логотипов в apk картинках
- Для детекта по pcap файлам трафика
- Для выявления аномалий в работе промышленных сетей
- …

Anti APT

- Анализ не только внешнего, но и внутреннего траффика
- Анализ потока событий на уровне системы в целом
- EDR (Endpoint Detection & Response): агенты и сервер
- SIEM (Security Information and Event Management): сбор и хранение (в удобном для работы виде) логов от разных приложений (в т.ч. от EDR и IDS/IPS).
- Threat Hunting: процесс цикличного поиска и устранения угроз

OpenIOC

```
<?xml version='1.0' encoding='UTF-8'?>
<ioc
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.mandiant.com/2010/ioc"
  id="e1cbf7ca-4938-4d3c-a7e6-3ff966516191"
  last-modified="2014-10-21T13:08:41Z">

  <definition>
    <Indicator id="e16e6299-f75b..." operator="OR">
         <IndicatorItem id="590-7df8..." condition="is">
           <Context document="PortItem"
                    search="PortItem/remoteIP" type="mir"/>
           <Content type="IP">70.85.221.10</Content>
         </IndicatorItem>

       <IndicatorItem id="5ea9f200-01f1..." condition="is">
         <Context document="FileItem"
                  search="FileItem/Md5sum" type="mir"/>
         <Content type="md5">8c4fa713c5e2b009114adda758adc445</Content>
       </IndicatorItem>

       <IndicatorItem id="3f83ca5b-9a2c..." condition="is">
         <Context document="ProcessItem"
                  search="ProcessItem/SectionList/MemorySection/Name"
                  type="mir"/>
         <Content type="string">Local Settings\Application Data\conhost.dll
         </Content>
       </IndicatorItem>

    </Indicator>
  </definition>
</ioc>
```

XML-схема для IOC

Проверяет наличие соединения системы с вредоносным IP-адресом

Проверяет наличие вредоносного файла с помощью контрольной суммы MD5

Проверяет наличие вредоносного процесса, запущенного локально

Индикаторы компрометации (IOC)

- Необычные DNS-запросы.
- Подозрительные файлы, приложения и процессы.
- IP-адреса и домены, принадлежащие ботнетам или командным серверам вредоносного ПО.
- Значительное количество обращений к одному файлу.
- Подозрительная активность в учетных записях администраторов или привилегированных пользователей.
- Неожиданное обновление программных продуктов.
- Передача данных через редко используемые порты.
- Нетипичное для человека поведение на веб-сайте.
- Сигнатура или хеш-сумма вредоносной программы.
- Необычный размер HTML-ответов.
- Несанкционированное изменение конфигурационных файлов, реестров или настроек устройства.
- Большое количество неудачных попыток входа в систему.

Threat intelligence

- Сбор и анализ информации об актуальных угрозах и группировках киберпреступников
- TIP (threat intelligence platform) – threat intelligence как сервис

Специализации

- Разработчики
- Data Science
- Malware analysts
- SOC (security operations center)
- CERT (computer emergency response team)
- Threat intelligence
- Pentest
- Application security
- Reverse engineering (включая anti-ransom)
- …

Computer Science

- Алгоритмическая эффективность
- Глубокое знание конкретных платформ и протоколов
- Языки программирования: разные
- Статистика, ML
- Ассемблер, компиляторы
- Криптография
- NLP
- …

Спасибо за внимание!

- dronkalegin@gmail.com
- https://vk.com/kalegin