

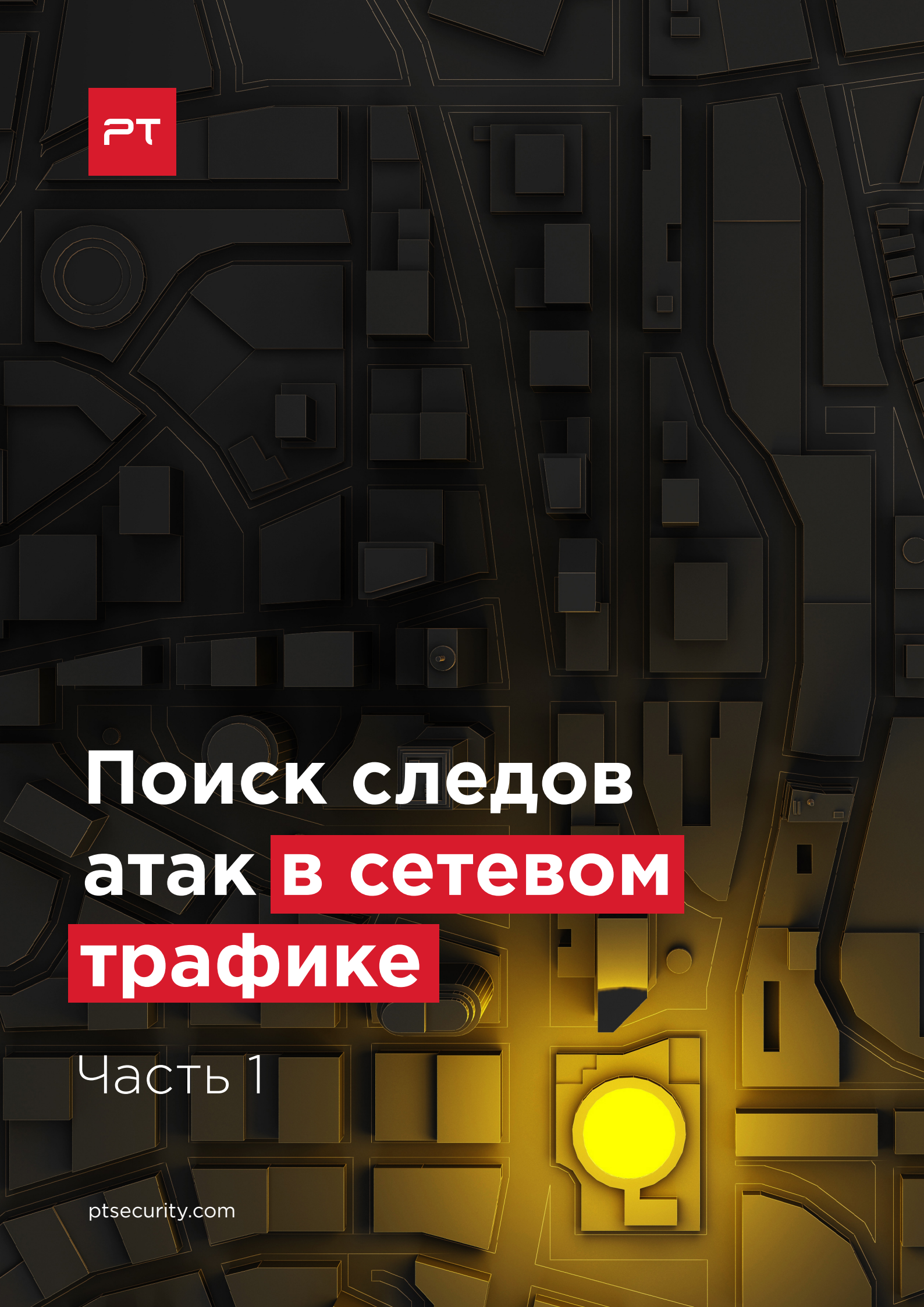


PT

# Поиск следов атак **в сетевом** **трафике**

Часть 2

ptsecurity.com



## Содержание

Анализируем подключения к командным серверам	2
Выявляем использование туннелей	2
Ловим вредоносные скрипты	3
Борьба с обфускацией	4
Как выявить атаку, если трафик зашифрован	6
Исследуем хакерский инструментарий	6
Заключение	9

Проводя целенаправленную атаку, нарушитель не может быть уверен, что после проникновения в локальную сеть организации он окажется в нужном ему сегменте сети. Для поиска ключевых серверов и рабочих станций потребуется разведка и ряд подключений между узлами. Такие подключения, или, как обычно говорят, *перемещение хакера внутри периметра*, — непременно оставят следы в сетевом трафике. Помимо подключений внутри сети злоумышленник должен установить связь с внешним командным сервером. Такие действия можно отследить, а значит, своевременно обнаружить кибератаку. Если сохранять копию трафика, анализ можно проводить и ретроспективно. Рассмотрим, как обнаруживать в трафике признаки использования некоторых популярных техник.

## Анализируем подключения к командным серверам

Вредоносные программы, которые оказались во внутренней сети организации, должны связываться со своими управляющими серверами, чтобы злоумышленники могли контролировать ход атаки. Основная задача при установке соединения — передавать данные в таком виде, который усложняет их обнаружение в общем потоке трафика. Существует множество методов сокрытия связи с командным сервером и усложнения анализа передаваемых данных. Это может быть, например, использование нестандартных алгоритмов кодирования, стеганографии или маскировка под легитимный трафик. Мы расскажем, как выявить в сетевом трафике подозрительные соединения, даже если они умело скрыты.

## Выявляем использование туннелей

Злоумышленники могут передавать вредоносный код внутри туннеля, установленного с использованием распространенного протокола, например DNS, SMTP, ICMP. Есть два подхода к выявлению туннелей: выявление признаков самого туннеля или индикаторов использования конкретной утилиты для его создания. В первом случае нужно знать, какие особенности в сетевом трафике указывают на наличие туннеля для каждого протокола. К примеру, для протокола DNS аномалией являются большие размеры TXT-записей. Признаком ICMP-туннеля может служить размер пакетов Echo Request и Echo Response: он лишь незначительно изменяется в зависимости от ОС, поэтому если пакет заметно больше обычного, это говорит об аномалии. Косвенным признаком может стать увеличение ICMP-трафика: обычно в



Чаще всего для обмена данными с командным центром или доступа к каким-либо внешним ресурсам злоумышленники используют широко распространенные протоколы прикладного уровня — HTTP, HTTPS, DNS; это помогает скрыть нелегитимный трафик в общем потоке. Поэтому в первую очередь необходимо уметь определять используемые протоколы передачи данных и разбирать их для извлечения данных.

сети его мало, а при передаче большого количества данных через туннель будет заметный всплеск.

Второй подход — обнаружение отдельных утилит. Например, использование инструментов ICMPTX и ICMPSH видно по особенностям ICMP-пакетов.

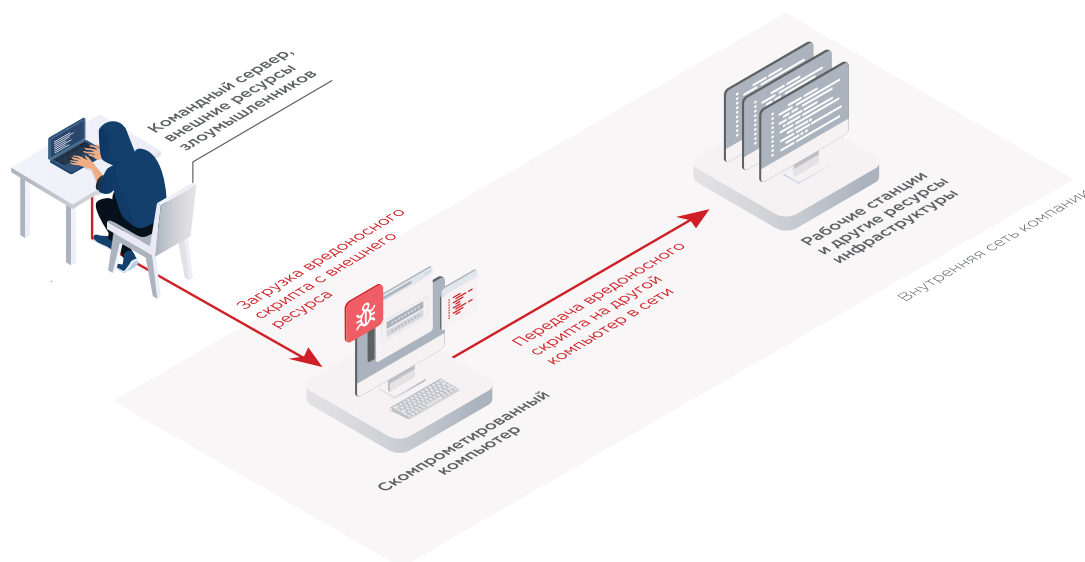
```

Domain Name System (response)
Transaction ID: 0xde41
> Flags: 0x8180 Standard query response, No error
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
▼ Queries
  8b2c0104e826c2838ee4b000022e707247. : type CNAME, class IN
    Name: 8b2c0104e826c2838ee4b000022e707247.
    [Name Length: 50]
    [Label Count: 3]
    Type: CNAME (Canonical NAME for an alias) (5)
    Class: IN (0x0001)
▼ Answers
  8b2c0104e826c2838ee4b000022e707247. : type CNAME, class IN, cname 39890104e825f1f2171b81ffff67aff188.
    Name: 8b2c0104e826c2838ee4b000022e707247.
    Type: CNAME (Canonical NAME for an alias) (5)
    Class: IN (0x0001)
    Time to live: 5
    Data length: 37
    CNAME: 39890104e825f1f2171b81ffff67aff188.
  
```

Рисунок 1. DNS-туннель

## Ловим вредоносные скрипты

В последнее время в атаках все чаще применяются скриптовые языки программирования. Перед тем как выполнить скрипт, его нужно передать на целевой узел. Это может быть не только файл скрипта с понятным для интерпретаторов расширением



(.ps1, .vbs, .bat или др.) или макрос внутри офисного документа. Преступники часто прибегают к другим методам, которые позволяют не оставлять лишних следов на узле. Тело скрипта может передаваться по сети, например, в виде ответа веб-сервера внутри HTML-кода, TXT-записи в DNS-ответе, закодированной строчки с командами на исполнение, передаваемой по протоколу WMI.



Для обнаружения разных способов передачи вредоносных скриптов необходимо уметь определять используемые протоколы и кодировку. Следует в автоматизированном режиме извлекать передаваемые данные, а найденные файлы отправлять на анализ в песочницу. Известные вредоносные утилиты можно также выявить по хеш-суммам, которые содержатся в специальных списках индикаторов компрометации.

```
HTTP/1.1 200 OK
Content-Length: 2417
Server: Microsoft-HTTPAPI/2.0
Date: Mon, 11 Nov 2019 10:13:41 GMT

<?XML version="1.0"?>
<scriptlet>
  <registration
    description="DebugShell"
    progid="DebugShell"
    version="1.00"
    classid="{90001111-0000-0000-0000-0000FEEDACDC}"
  >
    <script language="JScript">
      <![CDATA[
        while(true)
        {
          try
          {
            w = new ActiveXObject("WScript.Shell");
            h = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
            p = new ActiveXObject("WinHttp.WinHttpRequest.5.1");
            try
            {
              v = w.RegRead("HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Internet Settings\\ProxyServer");
              try
              {
                q = v.split("=")[1].split(";")[0];
                h.SetProxy(2,q);
                p.SetProxy(2,q);
              }
              catch(e)
              {
                h.SetProxy(2,v);
                p.SetProxy(2,v);
              }
            }
          }
        }
      ]>
    </script>
  </scriptlet>

```

Рисунок 2. Код утилиты JSRat, передаваемый в ответе веб-сервера

## Борьба с обфускацией

Перед злоумышленниками особенно остро стоит проблема обхода средств защиты, поскольку для успешного развития атаки внутри корпоративной сети необходимо оставаться незамеченными как можно дольше. Первым эшелоном выступают средства сигнатурного анализа трафика, потому что именно через них проходит сетевой трафик начальной компрометации и загрузки модулей на дальнейших стадиях.

Средства сигнатурного обнаружения выявляют уже известные угрозы. Такие угрозы исследуются аналитиками, которые выявляют общие признаки, на основании которых составляют правила и сигнатуры. В противовес такому типу обнаружения злоумышленники используют техники обфускации кода, кодирования и шифрования. Это либо разрушает искомый паттерн (обфускация), либо скрывает его от средства защиты (кодирование и шифрование).

Как правило, вредоносные программы используют те или иные методы кодирования информации. Наиболее распространенными являются Base-подобные кодировки, чаще всего это Base64. К примеру, ответ от агента Cobalt Strike содержит данные внутри POST-запроса, закодированные стандартной кодировкой Base64. При исследовании сетевого трафика нужно определять использование (как минимум) Base64 и применять правила анализа уже к декодированному содержимому.



```
function ntXZPl3vFDH2d() {
  Param(
    [Parameter(ValidateSet=1, Position=0)]
    [string] $MONvslfJ6doz
  )

  $SpPoJfXpeQcy += "TiqQAAVAAAAA/////
8AALgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA2AAAA4fug4tAnlIbgB7NWhGhpcy8wcn9ncfTIGNhsvdCB8IZ5Bdyd4gai4gRE9TIG1v2GUdQK9JAAAAAAACtHQI66VbmUQZtLpUGbSK
185eUqtZtLpUGStFbmIjF9rLmUG5vXNmBUNQZtLwvDS6FbmI3py2jpuG5AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUUAEEBQAKtYvCAAAAAAAdgAAB8CvEGABMAAAAGIAAGAAKBUAAEAAAAIAAAAAQAQAEAA
AALAAQAAAAAABAAAAAABAAAAAIAQAAQAAAAAAACEFAAAQAAQAAAAABAAABAAAAAQAAAAAAAAAAETHQAcraAAAAQhQhDYQQAAAAAAAAAAAAAAAAAAAAAAAAAAAA"

  $PgJvXpGcy += ...
}
```

Рисунок 3. Передача данных в кодировке Base64

```
Input                                     length: 512  
                                         lines:    1
```

```
TvqAAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA4fug4AtAnIbgBTM0hVghpcyBwcm9ncmFtIGQhbm5vdCB1ZSBydW4gaW4gRE9TIG1vZGUuDQoKJAAIAAAAAAAAACTMQlB6VBm0u1QztLpUGbSK1850utQZtLpUGStFTF8m0lpfo9LMUgb5vXNW0uNQztIuVMdS6FBm0eJpyYzjPUGbSAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAUUEUAEEwBBQAKfxVCIAAAAAAAAAAADGaA8BCwEGAAABmAAAAGIAAAgAAKU0AAAAEAAAAIAAAAAAAAAAAAAATATAAAQAAGAAAAABAAAAIAAAAAAAAAATAQAAQAIAAAAAAAAACAECFAAAQAAAQAAAAABAAAABAAAAIAAAQAAAAAAAEhQA0aoAAAAADQAwDYQQAAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Рисунок 4. Декодирование исполняемого файла

Одно из самых действенных средств борьбы с обфускацией — поведенческий анализ, поэтому хорошим решением станет объединение анализа сетевого трафика с запуском кода в специально созданной виртуальной среде (песочнице). Таким образом следует проверять все извлеченные из трафика файлы, например незашифрованные архивы или упакованные исполняемые файлы. Поведенческий анализ позволяет выявить подозрительные действия при выполнении кода и обнаружить новые, неизвестные ранее угрозы.

Нужно учитывать, что некоторые хакерские инструменты умеют определять, что выполнение происходит в виртуальной среде. В таком случае они не запускают вредоносные функции и ведут себя как легитимное ПО. Поэтому песочница должна принимать меры по сокрытию своего присутствия и распознавать техники обхода.



В случае сложных целенаправленных кампаний (APT-атак) соответствующие индикаторы компрометации могут быть еще неизвестны на момент проведения атаки. Поэтому важно проводить ретроспективный анализ файлов при появлении новых данных.

## Как выявить атаку, если трафик зашифрован

Есть разные подходы к детектированию подозрительной активности в зашифрованном трафике. Можно, например, расшифровывать трафик посредством атаки типа «человек посередине», однако использование нестандартных протоколов, а также SSL pinning (аутентификация клиента по сертификату при установлении SSL-соединения) вносят ограничения на применение активных методов анализа.

С другой стороны, существуют пассивные методы анализа; например, признаки вредоносной активности в зашифрованном трафике могут быть обнаружены через побочные каналы. Одним из таких методов является анализ длин пакетов и порядка их следования ([bit.ly/2USvWu2](https://bit.ly/2USvWu2)) с учетом закономерностей, выявленных аналитиками при исследовании определенных инструментов. Алгоритм работы вредоносной программы определен заранее. Агент на зараженном узле должен сообщить командному серверу свой идентификатор, наименование системы, в какой он запущен, и другую служебную информацию. Все это составляет внутренний протокол, или алгоритм взаимодействия клиента и сервера. И поскольку любой протокол заранее фиксирован, это зацепка для аналитика.

Итак, при заражении узла программа должна сообщить краткую информацию о нем. Такая информация при шифровании формирует пакеты определенной длины. Можно установить такие правила анализа длин запросов клиента и ответов сервера, чтобы однозначно идентифицировать определенное семейство ВПО. Точная длина начальных пакетов может варьироваться, но при задании достаточно узких рамок можно добиться баланса между true positive и false positive.

Важно отметить, что данный подход не зависит от того, какой криптографический протокол используется — стандартный, стандартный с модификациями или полностью самописный. Устранение побочных каналов должно осуществляться на уровне приложения, а не на уровне криптографического протокола, следовательно, пока создатели ВПО не начнут скрывать идентифицирующие их закономерности, метод будет успешно работать.

## Исследуем хакерский инструментарий

В большинстве своем киберпреступники используют уже готовые фреймворки, поэтому при поиске подозрительного трафика важно знать особенности, характерные для различных инструментов. Так, среди APT-группировок популярен фреймворк Koadic, который передает полезную нагрузку в виде ответа веб-сервера внутри HTML-кода. Сама полезная нагрузка при этом зашифрована, а к расшифровщику применены различные техники обфускации кода, в том числе случайные имена пользовательских функций, их аргументов и переменных. Зашифрованный скрипт дополнительно закодирован. Для обхода средств обнаружения вторжений (IDS) авторы скрывают имя функции, используемой для выполнения расшифрованного скрипта.

```

HTTP/1.0 200 OK
Server: Apache
Date: Thu, 15 Aug 2019 14:12:45 GMT

<html>
<head>
<script language="JScript">
window.moveTo(-1337, -2019);
window.blur();
window.resizeTo(2, 4);

try
{
    window.onerror = function(sMsg, sUrl, sLine) { return false; }
    window.onfocus = function() { window.blur(); }
}
catch (e){}

function ZRcLbGHkVHKgYE(habJjvflxNRatmsZqk,zblGxaqKSvaOaqBea){var
iqbiZdIedIJsNx='';while(zblGxaqKSvaOaqBea.length>habJjvflxNRa.msZqk.length){zblGxaqKSvaOaqBea+=zblGxaqKSvaOaqBea;}
for(i=0;i<habJjvflxNRatmsZqk.length;i++){var kfVNPnOkfXcKMZ=String.fromCharCode(parseInt(habJjvflxNRatmsZqk.substr(i,2),
16)+"zblGxaqKSvaOaqBea charCodeAt(i/2));iqbiZdIedIJsNx+=iqbiZdIedIJsNx+kfVNPnOkfXcKMZ;}
return iqbiZdIedIJsNx;}
var MRPTXbCrYldZuVv="}EcmiQJfhJmwvHIGXCTJXcLpK3bMwQcmFpYOLBceJqZaiqZJQmOvHrrcFvZVvBSlHngMIABlAlFuYcFMyzLufNdKlIdGdwJX";var
DJ3uTobxmNa="UxjCKPNNucDeBBdYulPkrJxfBhnxXbtIdIocuytGVMIlGBLvnXSWUyaJyXfDbjXAMBRhXhBNZQCrwZxtEYEOebUuyVxXNUddCCNYRvSvZpWzrciToC
E";var KrPCrTbBpLKYMhKpMu="TfgVnxYgqDeUOKDqvcfELLLtWoiAireCyszgkulelWVvYsFeqQzjTMkfngtEYLZsIRAcYQBRfJorNImlbXbojRoJETH";var
PtSDIjwGjZx="yGDCrOIMqIleKhXCDGNYXdiCmOXbMzAJaxCeokblPXhxbukCofPmyBIXLeHTOynUayZKfyInQYxrfLLfZZpElIOTHngLdgZAzUedaEwk";var
AlAnoqmqYeYELCaUu=[String.fromCharCode(MRPTXbCrYldZuVv.length),String.fromCharCode(DJ3uTobxmNa.length),String.fromCharCode(KrPCrT
bBpLKYMhKpMu.length),String.fromCharCode(PtSDIjwGjZx.length)];var lnhqTehmJofrbLoXHYFS=this[AlAnoqmqYeYELCaUu[0]]
+AlAnoqmqYeYELCaUu[1]+AlAnoqmqYeYELCaUu[2]
+AlAnoqmqYeYELCaUu[3]]+lnhqTehmJofrbLoXHYFS(ZRcLbGHkVHKgYE)'3d2e14471a102233b39190130296a29094d3d321d1a2d2e051c22385e2a1908273104
0720b304d4d252a11471631001f0615130004045231005e522328308f17233b1a115e36222303342e2100131d3f2925030423705d4d3d321d1a2d2e051c22385e9

```

Рисунок 5. Обфускация полезной нагрузки в Koadic

Ниже представлен ответ от агента Koadic командному серверу. Часть служебной информации, например тип исполненного задания, создатели хранят в собственных HTTP-заголовках. Идентификаторы сессии и задания задаются случайным образом, а путь к библиотеке mshtml обфусцирован, чтобы обойти сигнатуры IDS, но вид URI все равно является одним из индикаторов, по которому можно узнать Koadic.

[illegible]

Рисунок 6. Запрос, свидетельствующий об использовании Koadic

Опишем несколько подходов, которые помогают выявить зашифрованное соединение с использованием Meterpreter из состава фреймворка Metasploit.

Долгое время хорошо работало правило детектирования шелла Meterpreter reverse\_https, которое анализировало SSL-сертификат, с которым устанавливалось защищенное соединение. В сертификате, сгенерированном с помощью Metasploit, поля issuer и subject содержат идентичные наборы из шести relative distinguished name (RDN), расположенных в фиксированном порядке.

```
> issuer: rdnSequence (0)
  > rdnSequence: 6 items (pkcs-9-at-emailAddress=driver@simonis.reynolds.info,id-at-commonName=simonis.reynolds.info,
    > RDNSequence item: 1 item (id-at-countryName=US)
    > RDNSequence item: 1 item (id-at-stateOrProvinceName=NC)
    > RDNSequence item: 1 item (id-at-organizationName=Simonis-Reynolds)
    > RDNSequence item: 1 item (id-at-organizationalUnitName=driver)
    > RDNSequence item: 1 item (id-at-commonName=simonis.reynolds.info)
    > RDNSequence item: 1 item (pkcs-9-at-emailAddress=driver@simonis.reynolds.info)
  > validity
  > subject: rdnSequence (0)
    > rdnSequence: 6 items (pkcs-9-at-emailAddress=driver@simonis.reynolds.info,id-at-commonName=simonis.reynolds.info,
      > RDNSequence item: 1 item (id-at-countryName=US)
      > RDNSequence item: 1 item (id-at-stateOrProvinceName=NC)
      > RDNSequence item: 1 item (id-at-organizationName=Simonis-Reynolds)
      > RDNSequence item: 1 item (id-at-organizationalUnitName=driver)
      > RDNSequence item: 1 item (id-at-commonName=simonis.reynolds.info)
      > RDNSequence item: 1 item (pkcs-9-at-emailAddress=driver@simonis.reynolds.info)
    > subjectPublicKeyInfo
  > extensions: 2 items
```

Рисунок 7. Содержимое полей issuer и subject



Метод, основанный на выявлении сертификата, подходит для обнаружения простых вариантов шеллов с параметрами по умолчанию. Но в Metasploit есть возможность использовать сертификаты, созданные через другие утилиты, или имитировать сертификат легитимного ресурса ([bit.ly/2XloRDJ](http://bit.ly/2XloRDJ)). По результатам исследований специалисты PT Expert Security Center пришли к выводу, что на данный момент самый оптимальный подход — детектирование, основанное на длинах пакетов зашифрованного трафика.

Другим примером детектирования служит обнаружение работы Meterpreter reverse\_tcp. В начале соединения происходит отправка пакета определенной длины, внутри которого передается публичный ключ RSA-2048. Такой пакет дополнительно защищен шифром гаммирования (XOR), однако из-за малой длины гаммы в структуре пакета можно выявить повторяющиеся части. На рисунке ниже выделены подобные фрагменты, а также сам зашифрованный ключ.

0000	0e 56 45 34	0e 56 45 34	0e 56 45 34	0e 56 45 34	.VE4.VE4 .VE4.VE4
0010	0e 56 45 34	0e 56 45 34	0e 56 47 17	0e 56 45 34	.VE4.VE4 .VG..VE4
0020	0e 56 45 12	0e 57 45 35	6d 39 37 51	51 38 20 53	.VE..WE5 m97QQ8 S
0030	61 22 2c 55	7a 33 1a 40	62 20 1a 51	60 35 37 4d	a",Uz3.@ b .Q`57M
0040	7e 22 2c 5b	60 56 45 34	0e 7f 45 35	0e 54 7c 0c	~",[`VE4 ..E5.T .
0050	3a 60 72 01	3c 63 77 00	39 6e 72 01	3f 64 76 07	:`r.<cw. 9nr.?dv.
0060	3e 63 7c 03	38 64 71 02	37 65 76 02	3f 66 45 34	>c .8dq. 7ev.}fE4
0070	0e 57 89 34	0f 54 63 19	23 7b 68 19	4c 13 02 7d	.W.4.Tc. #{h.L..}
0080	40 76 15 61	4c 1a 0c 77	2e 1d 00 6d	23 7b 68 19	@v.aL..w ...m#{h.
0090	23 5c 08 7d	47 14 0c 5e	4f 18 07 53	65 27 2d 5f	#\..}G..^ O..Se'-_
00A0	67 11 7c 43	3e 14 04 65	4b 10 04 75	41 15 04 65	g. C>..e K..uA..e
00B0	36 17 08 7d	47 14 06 53	45 15 04 65	4b 17 70 77	6..}G..S E..ek.pw
00C0	6a 2f 23 65	49 23 23 77	3e 24 6a 07	5b 24 00 43	j/#eI##w >\$j.[\$.C
00D0	77 6f 4f 4c	7b 21 2a 7a	44 1f 23 58	7b 2c 20 65	woOL{!*z D.#X{, e
00E0	6d 11 77 77	66 3a 75 5a	69 62 07 46	45 17 0c 0c	m.wwf:uZ ib.FE...
00F0	45 7d 26 5c	57 1c 28 4c	63 6e 32 42	43 67 7d 7d	E}&\W.(L cn2BCg}}}
0100	7f 15 0a 67	4d 06 0a 5a	58 17 04 7a	69 11 24 5a	...gM..Z X..zi.\$Z
0110	62 18 6e 3e	4f 00 6a 76	41 63 33 76	4a 1d 37 42	b.n>O.jv Ac3vJ.7B
0120	40 11 77 52	7f 62 75 65	42 6e 0a 70	63 21 01 71	@.wR.bue Bn.pc!.q
0130	54 3c 20 58	46 62 04 73	3a 27 76 02	5d 12 6a 03	T< XFb.s :`v.].j.
0140	36 39 24 66	7f 1e 04 42	5c 03 11 4e	41 1f 0c 56	69\$f...B \..NA..V
0150	74 1c 16 1f	04 0f 32 46	58 1c 20 04	48 04 36 00	t.....2F X. .H.6.
0160	45 02 36 46	79 6e 09 65	3f 3f 06 65	3e 65 1c 02	E.6Fyn.e ??>e>e..
0170	43 20 2c 77	64 24 2f 55	7e 21 13 71	39 12 06 50	C ,wd\$/U ~!.q9..P
0180	6d 33 32 0d	62 19 7d 53	78 14 31 64	48 14 34 1f	m32.b.}S x.1dH.4.
0190	39 1f 26 42	7b 5c 24 7e	40 66 36 50	63 21 6a 70	9.&B{\'\$~ @f6Pc!jp
01A0	4f 3c 23 40	38 7d 16 71	5c 6f 2e 70	77 3d 10 51	O<#@8}.q \o.pw=.Q
01B0	5c 2f 35 65	5a 65 27 03	4f 15 7c 5f	4b 0e 0c 1f	\5eZe'. O. _K...
01C0	56 23 04 4d	44 6e 02 7f	3d 0f 0c 78	57 64 11 61	V#.MDn.. =..xWd.a
01D0	4a 27 71 60	48 1b 4f 4d	3d 18 13 43	62 3f 20 78	J'q`H.OM =..Cb? x
01E0	49 0f 32 67	67 04 03 59	47 01 1d 45	4b 1f 16 64	I.2gg..Y G..EK..d
01F0	43 34 22 50	5d 1c 04 6d	6b 64 31 55	4d 64 71 07	C4"P]..m kd1UMdq.
0200	3f 30 17 6d	21 22 37 0d	7f 67 24 1f	5b 07 71 57	?0.m!"7. .g\$.[.qW
0210	7d 11 75 5f	6a 62 2f 3e	42 21 0c 70	4f 07 04 76	.}u_jb/> B!.pO..v
0220	04 7b 68 19	23 7b 00 7a	4a 76 15 61	4c 1a 0c 77	.{h.#{.z Jv.aL..w
0230	2e 1d 00 6d	23 7b 68 19	23 5c 45		...m#{h. #\E

Рисунок 8. Передача публичного ключа при установке соединения через Meterpreter reverse\_tcp

Благодаря большому количеству нулей в начале шифруемых данных и малой длине гаммы можно без труда найти ключ и посмотреть, что находится за XOR.

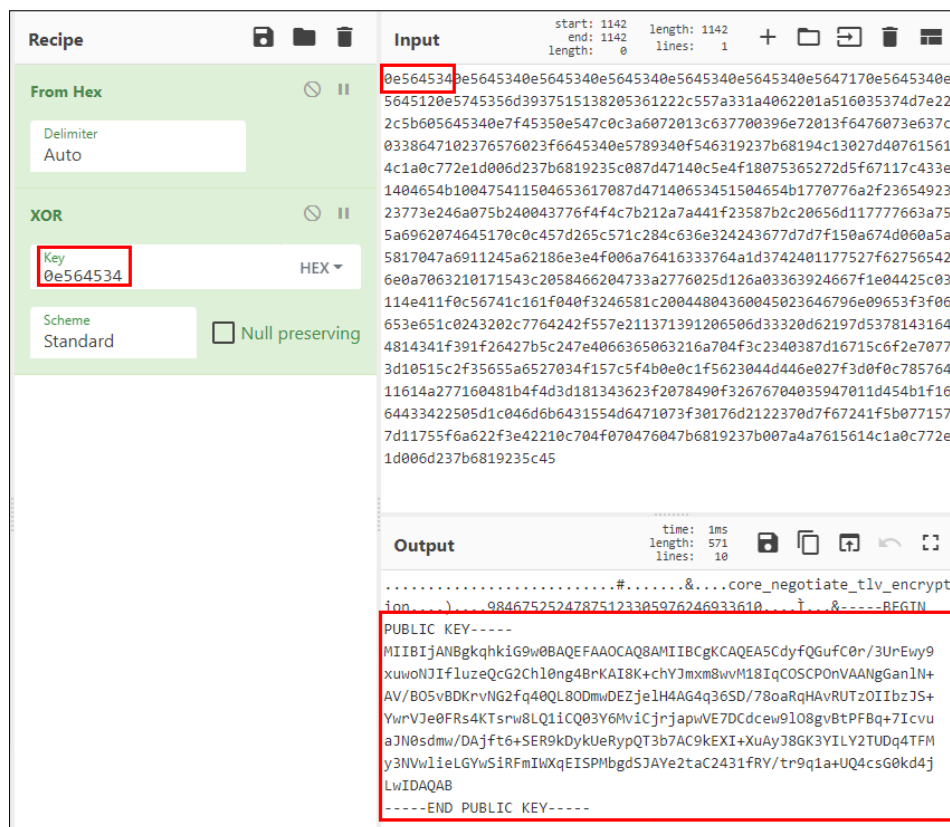


Рисунок 9. Расшифровка публичного ключа

Для фреймворка Cobalt Strike также были выявлены неизменные паттерны, связанные с особенностями используемого SSL-сертификата, которые позволяют обнаружить скрытую коммуникацию.

## Заключение

Хакеры постоянно совершенствуют свои инструменты, они используют самые новые образцы вредоносного ПО — только что созданные с помощью эксплойт-билдеров, обфусцированные, упакованные. Под каждую организацию может создаваться отдельный образец вредоноса, чтобы обойти традиционные средства обнаружения. В таком случае сигнатурная защита может не сработать, а вот анализ трафика, особенно ретроспективный, более эффективен: каждый раз переписывать сетевой протокол гораздо труднее, ведь надо модифицировать не только код клиента, но и подстраивать обработчик на командном сервере.

Анализ сетевого трафика может служить дополнением к уже существующим средствам обнаружения атак. Копия сетевого трафика позволяет восстановить последовательность действий злоумышленников — проанализировать взаимодействия между узлами сети, подключения к внешним ресурсам, командным серверам и детально разобраться в произошедшем инциденте. Вместе с проверкой передаваемых по сети файлов в песочнице такой подход дает возможность обнаружить даже сложную APT-атаку.

---

## О компании

[ptsecurity.com](http://ptsecurity.com)  
[pt@ptsecurity.com](mailto:pt@ptsecurity.com)  
[facebook.com/PositiveTechnologies](https://facebook.com/PositiveTechnologies)  
[facebook.com/PHDays](https://facebook.com/PHDays)

Positive Technologies уже 18 лет создает инновационные решения в сфере информационной безопасности. Продукты и сервисы компании позволяют выявлять, верифицировать и нейтрализовать реальные бизнес-риски, которые могут возникать в IT-инфраструктуре предприятий. Наши технологии построены на многолетнем исследовательском опыте и экспертизе ведущих специалистов по кибербезопасности.

Сегодня свою безопасность нам доверяют более 2000 компаний в 30 странах мира. В числе наших клиентов в России — 80% участников рейтинга «Эксперт-400».

Следите за нами в соцсетях ([Facebook](#), [ВКонтакте](#), [Twitter](#)), а также в разделе «[Новости](#)» на сайте [ptsecurity.com](http://ptsecurity.com).

---