?????? NetFlow

Выявляет необычные паттерны трафика, которые могут указывать на вредоносную активность. Дамп трафика содержит более детальную информацию, которая может быть проанализирована для выявления конкретных векторов атак и методов, используемых злоумышленниками. Техника направлена на более глубокое понимание сетевой активности для предотвращения атак.

Анализ сетевого трафика

Для выявления аномалий необходимо иметь возможность составить набор из типичного трафика, который в итоге образует так называемый "базовый уровень", на фоне которого будут определяться аномалии.

Базовый уровень — набор временных данных, который используется для сравнения с новыми данными с целью выявления аномалий или аномальных паттернов.

Злоумышленники скрывают присутствие в инфраструктуре. В трафике это видно как аномальное поведение. В каких случаях используют анализ сетевого трафика:

- Сбор трафика в реальном времени для выявления угроз (сигнатурный анализ, правила корреляции).
- Фиксация обычных для инфраструктуры внутренних взаимодействий в сети.
- Идентификация и анализ трафика на нестандартных портах и/или новых в сети хостах.
- Обнаружение проблем в сети или в веб-приложениях.
- Обнаружение ВПО.
- Активный поиск угроз (Threat hunting).

Например, всплеск SYN-пакетов, отправленных на множество различных портов — это с огромной вероятностью будет свидетельствовать о вертикальном сканировании.

TCP: Механизм тройного рукопожатия в TCP

Тройное рукопожатие используется в TCP для установления соединения между двумя хостами (клиентом и сервером):

1. Отправка запроса на соединение (SYN): Когда клиенту нужно установить соединение с сервером, он отправляет пакет с флагом SYN (synchronize) серверу
2. Подтверждение запроса и отправка запроса на соединение обратно (SYN-ACK): Сервер, получив пакет с флагом SYN, отвечает на него, устанавливая флаги SYN и ACK (acknowledgment). Флаг ACK указывает, что сервер получил пакет SYN от клиента. Этот пакет с флагами SYN и ACK отправляется обратно клиенту

3. Подтверждение ответа сервера (ACK): Клиент, получив пакет с флагами SYN и ACK от сервера, отправляет ответный пакет с флагом ACK. Этот пакет подтверждает, что клиент успешно получил подтверждение от сервера.

После завершения этой процедуры обе стороны могут начать обмен данными. Теперь соединение установлено и готово к передаче информации в обоих направлениях.

Нормальным окончанием сессии считается ситуация, когда клиент и сервер обмениваются пакетами с флагами FIN и ACK. Одна сторона отправляет FIN, вторая ответом отправляет ACK и FIN, вторая отправляет ACK.

О "ненормальном" закрытии сессии свидетельствуют пакеты с флагами RST (reset).Пакет с флагом RST закрывает сессию, не дожидаясь обмена пакетами с флагом FIN между хостами.

Методы HTTP

Для выполнения операций вроде загрузки страницы, запроса о скачивании файла или публикации чего-либо на сайт используются определенные методы. Эти методы определяют действия, совершаемые при запросе URI.

Методы GET и HEAD всегда должны работать в стандартной имплементации HTTP. Другие методы являются опциональными функциями, которые может разрешить владелец ресурса. Примером этого может быть доступная только для чтения страница вроде поста в блоге. Клиент может запросить от нее ресурсы и данные, но не способен модифицировать, добавлять или удалять их. Методы:

- **GET** — наиболее распространенный метод, запрашивающий информацию и контент с сервера. Например, GET http://10.1.1.1/Webserver/index.html затребует страницу index.html с сервера согласно представленному URI.
- **HEAD** — безопасный метод, требующий от сервера ответа схожего с запросом GET но без включения тела сообщения. Метод помогает получить информацию о сервере и его статусе.
- **POST** — способ отправить информацию на сервер, основываясь на заполненных полях в запросе. Например, отправление сообщения в Facebook или на форуме сайта это действие POST. Конкретное действие может отличаться в зависимости от сервера и поэтому следует обращать внимание на ответные коды подтверждения.
- **PUT** — метод использует приложенные к сообщению данные и помещает их по запрошенному URI. Если такого предмета еще не существует, то он будет создан из приложенных данных. Если он уже существует, то новый PUT будет считаться обновлением и предмет будет модифицирован соответствующим образом. Наиболее просто иллюстрирует различие между PUT и POST следующее: PUT создает или обновляет объект по указанному URI, в то время как POST создает дочерние сущности по соответствующему URI. Его можно также сравнить с разницей между созданием нового файла и комментарием, оставленным на странице с файлом.
- **DELETE** — удаляет предмет по указанному URI.

- **TRACE** — позволяет произвести удаленную диагностику сервера. При использовании метода TRACE удаленный сервер ответит тем же запросом, который был ему отправлен.
- **OPTIONS** - метод, позволяющий собрать информацию о поддерживаемых сервером методах HTTP. Таким образом мы можем определить требования к взаимодействую с определенным ресурсом или сервером без собственно запросов от него объектов или данных.
- **CONNECT** - метод, отведенный для использования с прокси и другими инструментами безопасности вроде межсетевых экранов. CONNECT позволяет туннелировать через HTTP (SSL туннели).

Коды ответа HTTP

Коды ответа HTTP (HTTP status codes) представляют собой числовые значения, которые отправляются сервером в ответ на запрос клиента. Эти коды позволяют клиентским приложениям понимать результат запроса и принимать соответствующие действия. Коды ответа HTTP делятся на пять основных классов:

1xx (Informational): используются для информационных сообщений	100 Continue: Сервер готов продолжить выполнение запроса клиента после того, как клиент отправит заголовок Expect. 101 Switching Protocols: Сервер согласен изменить протоколы по запросу клиента.
2xx (Successful): сообщают об успешном выполнении запроса клиента	200 OK: Запрос успешно выполнен. 201 Created: Запрос успешно выполнен, и ресурс был создан. 204 No Content: Запрос успешно выполнен, но в ответе нет содержимого.
3xx (Redirection): указывают, что клиент должен выполнить дополнительные действия для завершения запроса	301 Moved Permanently: Ресурс перемещен по новому URL и клиент должен использовать новый URL для всех последующих запросов. 302 Found (or Moved Temporarily): Ресурс временно перемещен. Клиент должен использовать новый URL, но старый URL может быть в будущем восстановлен. 304 Not Modified: Ресурс не изменился с момента последнего запроса клиента. Клиент может использовать кэшированную версию ресурса.
4xx (Client Error): Ошибки на стороне клиента.	400 Bad Request: Некорректный запрос со стороны клиента. 401 Unauthorized: Клиент не авторизован и требуется аутентификация. 403 Forbidden: Клиенту запрещен доступ к запрашиваемому ресурсу. 404 Not Found: Запрашиваемый ресурс не найден на сервере.

5xx (Server Error): Ошибки на стороне сервера.	500 Internal Server Error: Общая ошибка сервера. Этот код возвращается, если сервер не может определить природу ошибки. 502 Bad Gateway: Сервер, выступая в роли шлюза или прокси, получил недопустимый ответ от верхнего уровня сервера. 503 Service Unavailable: Сервер временно не может обрабатывать запросы. Это может быть из-за перегрузки сервера или его технических проблем.

Заголовки HTTP

Заголовки содержат ключевую информацию о данных, которые передаются, формате сообщения, авторизации, кэшировании и других аспектах HTTP-протокола.

Рассмотрим некоторые распространенные заголовки HTTP:

1. Общие заголовки (Common Headers):
   - Cache-Control: Управляет кэшированием, указывая, должны ли данные кэшироваться и как долго.
   - Date: Содержит дату и время, когда сообщение было отправлено.
   - Connection: Управляет соединением между клиентом и сервером, указывая, должно ли соединение быть закрытым после завершения запроса/ответа.
2. Заголовки запроса (Request Headers):
   - Host: Содержит доменное имя сервера и порт, к которому выполняется запрос.
   - User-Agent: Идентифицирует программное обеспечение, инициировавшее запрос (например, браузер или мобильное приложение).
   - Accept: Указывает типы медиа-ресурсов, которые клиент может принимать.
   - Authorization: Используется для передачи информации об аутентификации для доступа к защищенным ресурсам.
3. Заголовки ответа (Response Headers):
   - Location: Используется для перенаправления клиента на другой ресурс или URL.
   - Server: Содержит информацию о веб-сервере, который обрабатывает запрос.
   - Content-Type: Указывает тип медиа-ресурса в теле ответа (например, текст, изображение, JSON и т. д.).
   - Set-Cookie: Устанавливает куки на стороне клиента, позволяя серверу хранить информацию о состоянии сеанса.
4. Заголовки сущности (Entity Headers):
   - Content-Length: Указывает размер тела сообщения в октетах (байтах).
   - Content-Encoding: Указывает метод сжатия данных, используемый в теле сообщения.
   - Content-Disposition: Позволяет указать, должно ли содержимое быть отображено в браузере или загружено как файл.

TLS-рукопожатие

1. Приветствие (ClientHello): Процесс начинается с того, что клиент отправляет серверу сообщение ClientHello, в котором он указывает поддерживаемые криптографические алгоритмы и другие параметры.

2. Ответ сервера (ServerHello): Сервер выбирает подходящие параметры из списка, предложенного клиентом, и отправляет сообщение ServerHello в ответ. Это сообщение также включает в себя сертификат сервера, который клиент может использовать для проверки подлинности сервера.

3. Аутентификация сервера (Server Certificate): Сервер отправляет свой цифровой сертификат клиенту, который содержит публичный ключ сервера и информацию о сертификационном центре, который подписал сертификат.

4. Аутентификация клиента (Optional Client Certificate): Если сервер требует аутентификацию клиента, он может запросить клиентский сертификат. Клиент отправляет свой сертификат серверу, который также содержит публичный ключ клиента.

5. Обмен ключами (Key Exchange): Клиент и сервер обмениваются ключами для шифрования и расшифрования данных. Это может включать в себя процесс Diffie-Hellman обмена ключами или использование предварительно распределенных ключей (Pre-Shared Key) в случае TLS 1.3.

6. Завершение рукопожатия (Finished): После установки ключей и параметров шифрования, клиент и сервер отправляют Finished сообщения друг другу для подтверждения, что рукопожатие завершено успешно.

7. После завершения TLS рукопожатия оба конца соединения могут безопасно обмениваться данными, которые автоматически шифруются и расшифровываются с использованием установленных ключей.

SNI

SNI (Server Name Indication) — это расширение протокола TLS. SNI позволяет клиентскому устройству указывать серверу, с каким конкретным доменным именем оно хочет установить защищенное TLS-соединение. Это особенно важно в сценариях виртуального хостинга, где на одном сервере размещаются несколько веб-сайтов с разными доменными именами.

Анализ SNI в HTTPS трафике проводится для следующих целей:

- определение, к каким конкретным ресурсам обращается хост;
- управление доступом к ресурсам в зависимости от запрошенных доменных имен;
- обнаружение вредоносных или подозрительных доменных имен.

Протоколы прикладного уровня

FTP

FTP по своей сути является небезопасным протоколом и большинство пользователей для передачи данных через безопасный канал используют такие инструменты как SFTP.

FTP использует сразу несколько портов одновременно. FTP использует порты 20 и 21 TCP. Порт 20 используется для передачи данных, порт 21 используется для исполнения управляющих FTP-сессией команд.

FTP может работать в двух режимах. Активный это обычный операционный метод FTP, означающий что сервер ожидает от клиента контрольной команды PORT, заявляющей используемый для передачи данных порт. Пассивный режим открывает доступ к FTP-серверам, находящимся за межсетевым экраном или запрещающим прямое TCP-подключение NAT. В таком случае клиент посылает команду PASV и ждет отклика сервера, информирующего клиента о используемых для передачи данных IP и порте.

Команды, использующие порт 21:

- USER указывает на логин пользователя;
- PASS отправляет пароль пытающегося залогиниться пользователя;
- PORT в активном режиме изменяет используемый данными порт;
- LIST отображает список файлов в текущей директории;
- CWD изменяет текущую рабочую директорию на указанную;
- SIZE возвращает размер указанного файла;
- RETR скачивает файл с FTP-сервера;
- QUIT заканчивает сессию.

SMB

SMB (Server Message Block) используется для обмена файлами, принтерами, портами и другими ресурсами, чаще всего используется Microsoft Windows. SMB ориентирован на соединение протоколом, требует аутентификации пользователя от хоста к ресурсу для подтверждения допуска. В прошлом SMB использовал в качестве транспортного механизма NetBIOS через порты UDP 137 и 138. После современных изменений SMB также поддерживает прямой TCP-транспорт через порт 445, NetBIOS через TCP порт 139 и даже протокол QUIC.

Большое количество повторяющихся неуспешных логинов может означать о попытке получить доступ к аккаунту пользователя или использовать его права. Это распространенная тактика, использующая украденные у аутентифицированного пользователя учетные данные и права для горизонтального перемещения по системе или получения доступа к ресурсам.

DNS

Протокол DNS (Domain Name System) отвечает за преобразование доменных имен (например, www.example.com) в IP-адреса, которые используются для обмена данными в сети.

Записи DNS:

A (Address) Record: Сопоставляет доменное имя с IPv4-адресом.
AAAA (IPv6 Address) Record: Сопоставляет доменное имя с IPv6-адресом.
CNAME (Canonical Name) Record: Устанавливает альтернативное доменное имя для существующего доменного имени.
MX (Mail Exchange) Record: Указывает почтовый сервер, который обрабатывает почтовые сообщения для доменного имени.
NS (Name Server) Record: Определяет DNS-серверы, ответственные за зону доменного имени.
PTR (Pointer) Record: Используется для обратного разрешения и сопоставляет IP-адрес с доменным именем.
TXT (Text) Record: Позволяет добавить произвольный текст к записи DNS.

ICMP-туннель — это метод, используемый злоумышленниками для передачи данных через сеть, используя ICMP-пакеты вместо стандартных данных. Туннелирование помогает обойти правила IDS/IPS и/или межсетевых экранов, так как трафик конкретного приложения (например, ssh) передается под видом ICMP пакетов.

Детектирование ICMP-туннелей:

- Мониторинг сетевого трафика и поиск необычных или больших объемов ICMP-пакетов может помочь выявить использование ICMP-туннелей.
- Использование IDS/IPS/NTA для выявления аномалий в трафике, включая необычные или слишком частые запросы ICMP, может быть полезным инструментом для обнаружения таких атак.
- Анализ содержимого ICMP-пакетов, включая данные и структуру, может помочь выявить ICMP-туннелирование. Это может быть сделано с использованием специализированных инструментов и программных средств анализа сети типа NTA.

Детектирование DNS-туннелей:

- Анализ трафика: трафик DNS-туннелей на первый взгляд будет мало отличаться от обычного DNS трафика, особенно если у нас нет информации о конкретном хосте, и трафик снимался просто по 53 порту - на фоне обычного трафика туннель может "потеряться" из вида, но зная методы обнаружения DNS-туннелей, их будет возможно обнаружить в дампе.
- Обнаружение аномалий в трафике: существуют сигнатуры IDS/IPS для детектирования туннелей, они могут быть полезным инструментом для обнаружения DNS-туннелирования.
- Мониторинг DNS-логов: Мониторинг и анализ DNS-логов может помочь выявить необычные запросы и ответы, которые могут свидетельствовать о DNS-туннелировании.

NetFlow

Разработана Cisco для мониторинга и сбора статистики сетевого трафика. NetFlow собирает информацию о трафике в виде потоков на уровне сетевого устройства.

Каждое устройство, поддерживающее NetFlow, генерирует записи о трафике, содержащие информацию о переданных пакетах, объеме данных, источнике, назначении, протоколе и других параметрах.

Собранные данные могут быть использованы для анализа трафика и выявления различных характеристик сетевой активности. Это может включать в себя определение причин задержек в сети, выявление аномалий в трафике, в том числе NetFlow позволяет обнаруживать вредоносный трафик - например, DDoS атаки и сканирования, а также многое другое.

NetFlow включает в себя следующие компоненты:

- **NetFlow exporter (сенсор)** — устройство, поддерживающее технологию NetFlow, например, маршрутизатор, работает в качестве сенсора и собирает информацию о трафике. Оно агрегирует пакеты данных в потоки и экспортирует записи NetFlow по протоколу UDP на один или несколько коллекторов NetFlow. Поток, распознаваемый сенсором, должен иметь по крайней мере один из следующих общих параметров: порт входного интерфейса, IP-адрес и порты источника, и назначения и протокол 3 уровня. Поток считается готовым для экспорта в NetFlow, когда он неактивен в течение заданного периода времени или когда флаг TCP (например, FIN или RST) указывает на завершение потока.
- **NetFlow collector (коллектор)** — может быть на базе аппаратных или программных средств, однако программные средства используются более часто. Коллекторы NetFlow получают данные потоков от сенсоров, предварительно их обрабатывают и сохраняют в хранилище.
- **NetFlow analyzer (анализатор)** — обрабатывает и анализирует потоки, полученные и сохраненные коллектором. Он преобразует данные в отчеты и алерты, предоставляя в них информацию о характеристиках трафика, которые могут выявлять угрозы безопасности и проблемы в трафике.

NetFlow использует UDP или SCTP (Stream Control Transmission Protocol) для передачи данных от сенсора до коллектора. Как правило, коллектор слушает порты 2055, 9555 или 9995.

Потоки содержат в себе следующие поля в зависимости от протокола 3 уровня:

- номер версии протокола;
- номер записи;
- входящий и исходящий сетевой интерфейс;
- время начала и конца потока;
- количество байт и пакетов в потоке;
- IP адрес источника и назначения;
- порт источника и назначения;
- номер протокола IP;
- значение Type of Service;
- флаги TCP-соединений;
- адрес шлюза;

- маски подсети источника и назначения.

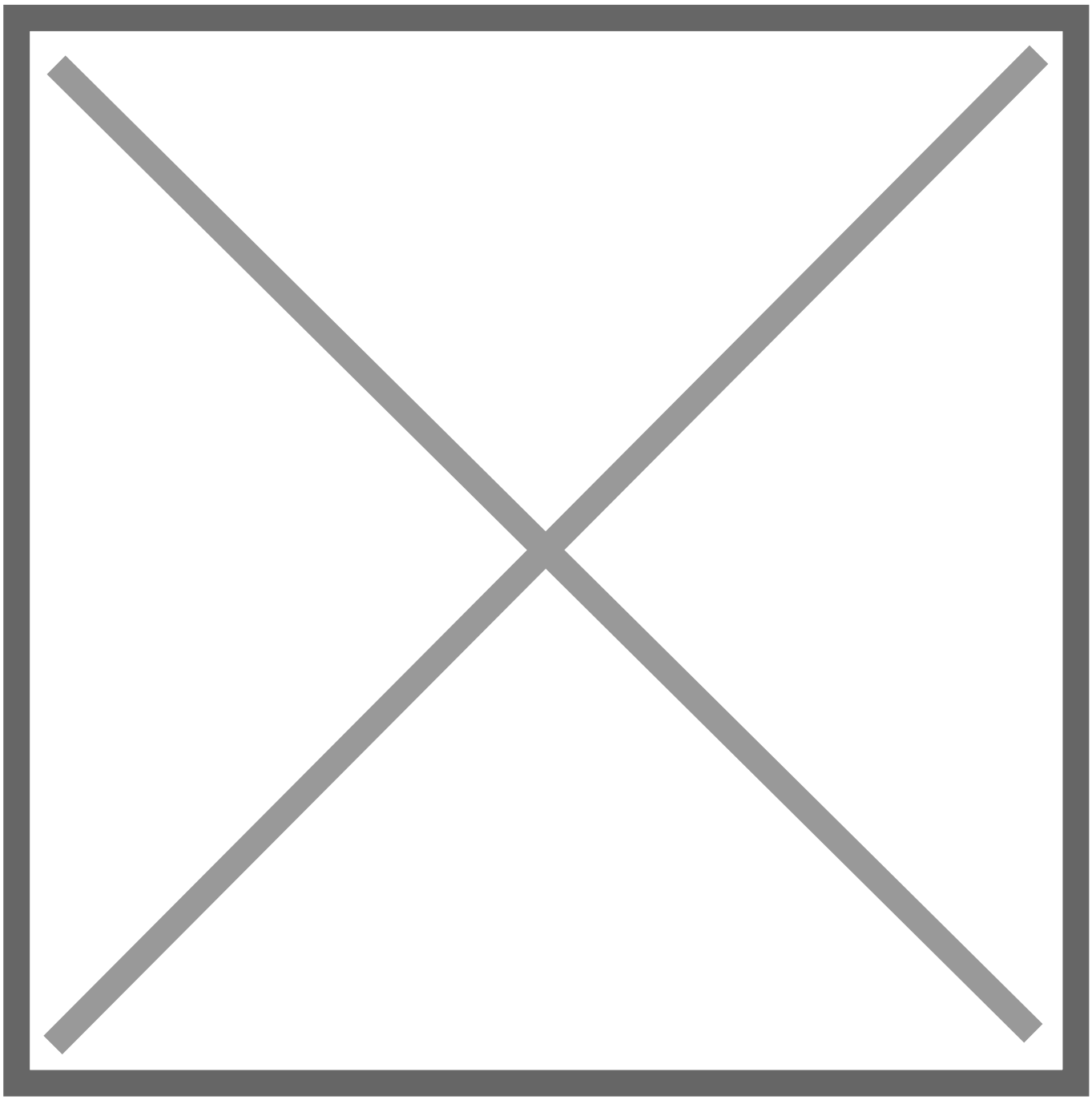
Обнаружение сканирования портов с помощью NetFlow

Предположим, было запущено простое сканирование открытых портов с помощью сканера портов Nmap с аргументами

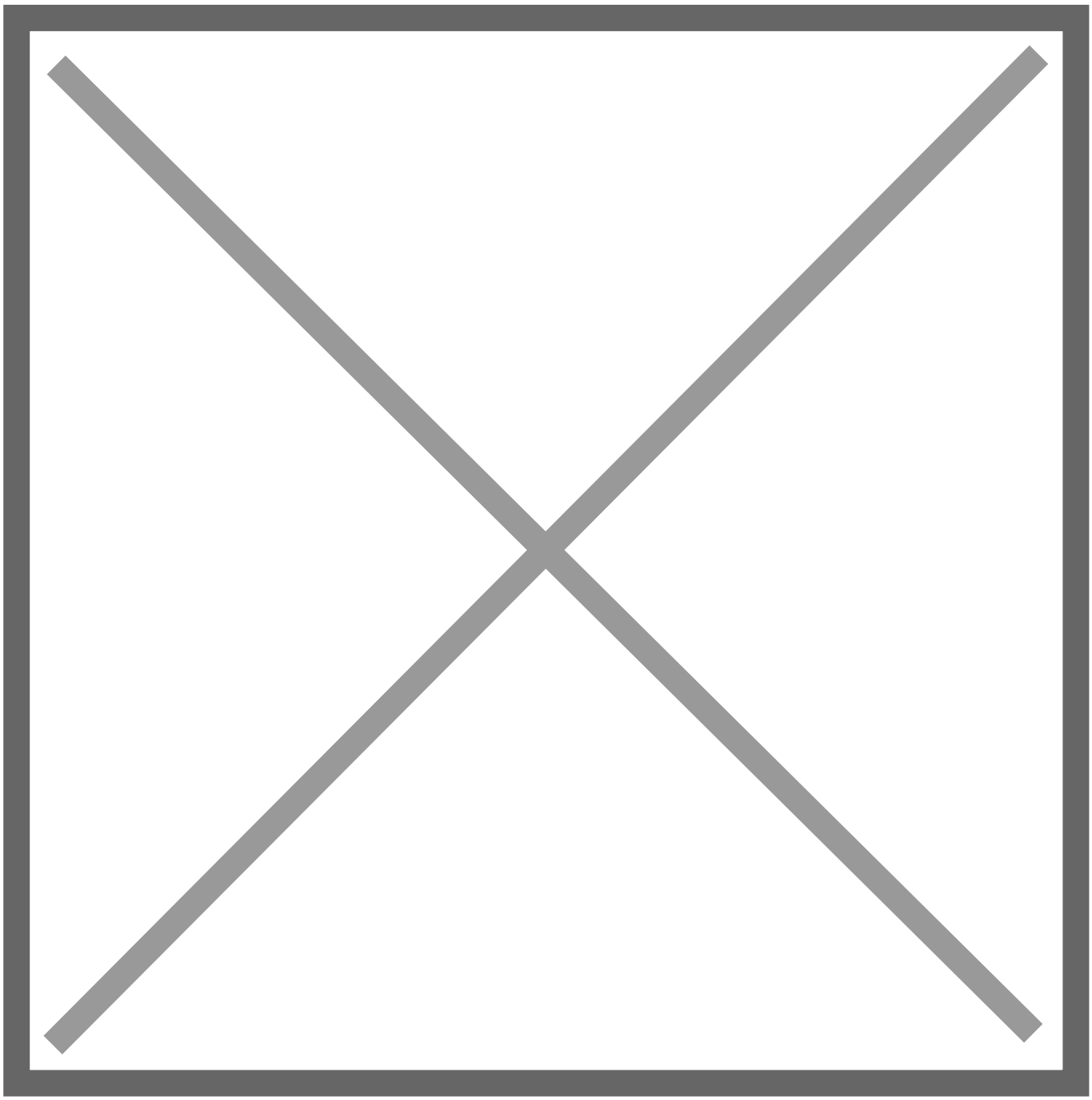
```
nmap -Pn -sV 192.168.1.2
```

Рассмотрим как сканирование открытых портов будет журналировано и задетектировано с помощью решения NetFlowAnalyzer от ManageEngine (данное решение является коммерческим с возможностью использования в течении 30дневного пробного периода и выбран в целях демонстрации в рамках данного курса).

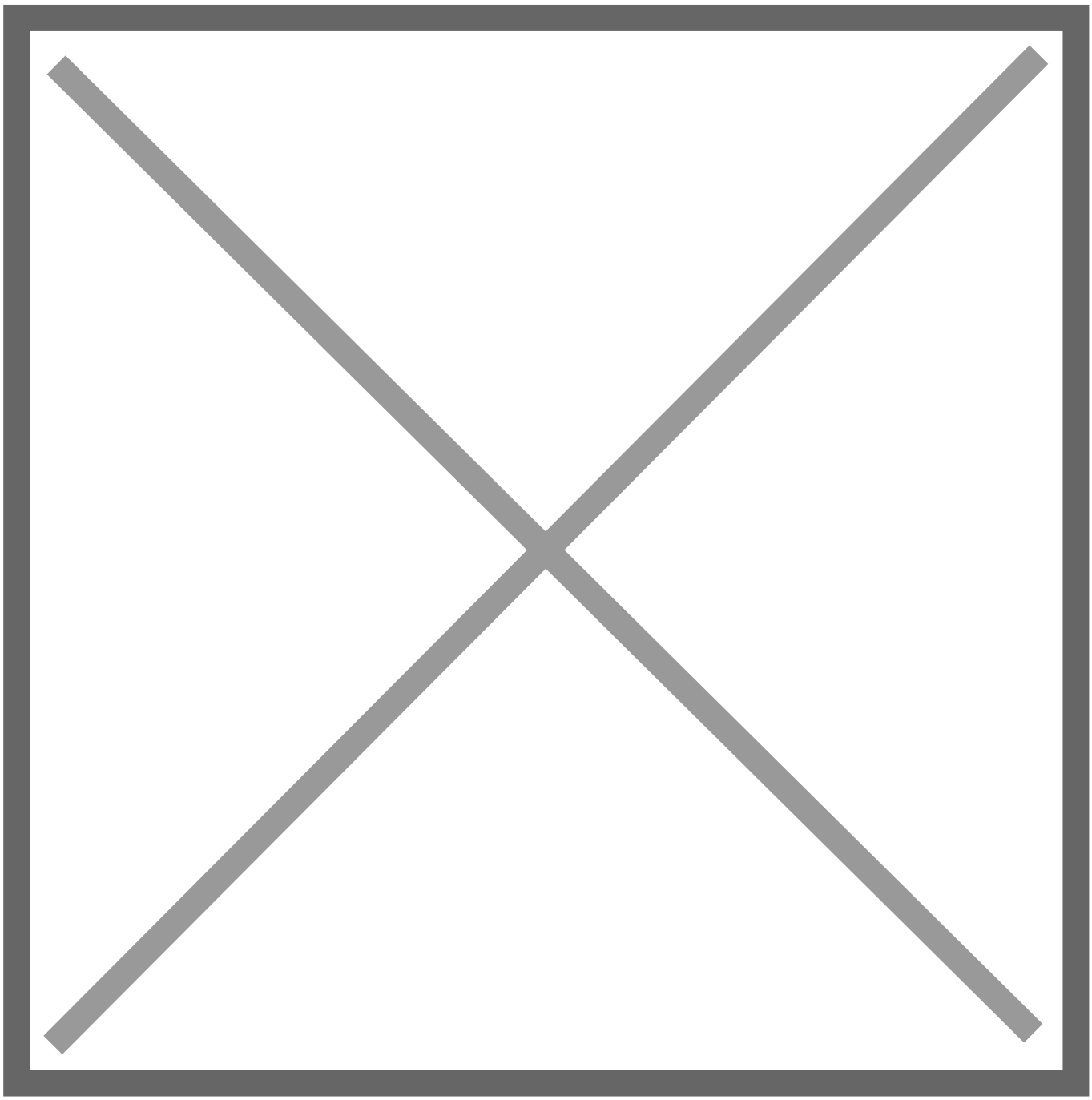
Если посмотреть на скриншот ниже, то можно увидеть резкий всплеск сетевого трафика в 17:55 направленного на хост с IP-адресом 192.168.1.2.



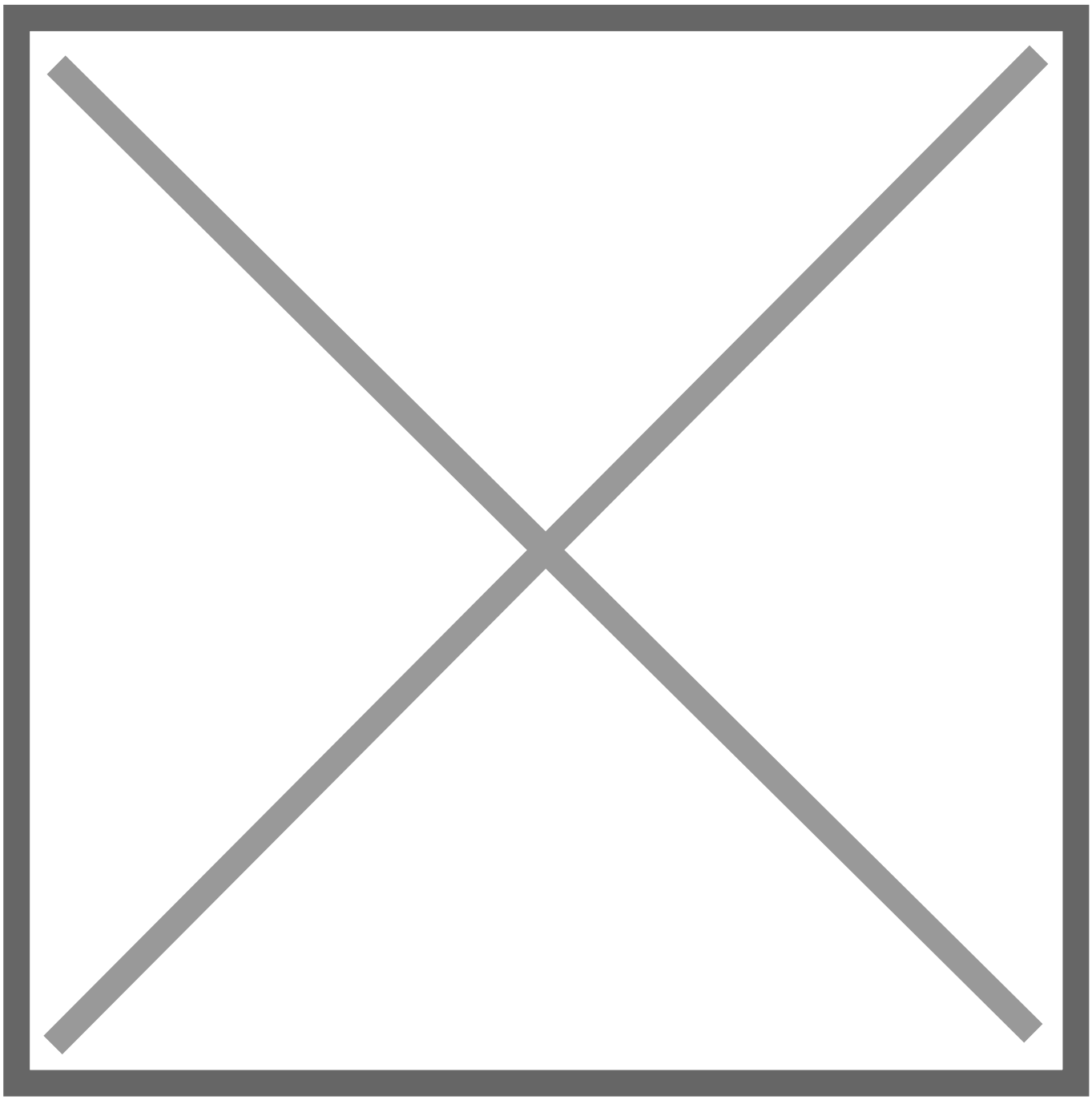
На скришоте ниже видно количество входящих сетевых пакетов, которые в момент сканирования портов достигли значения 2116.



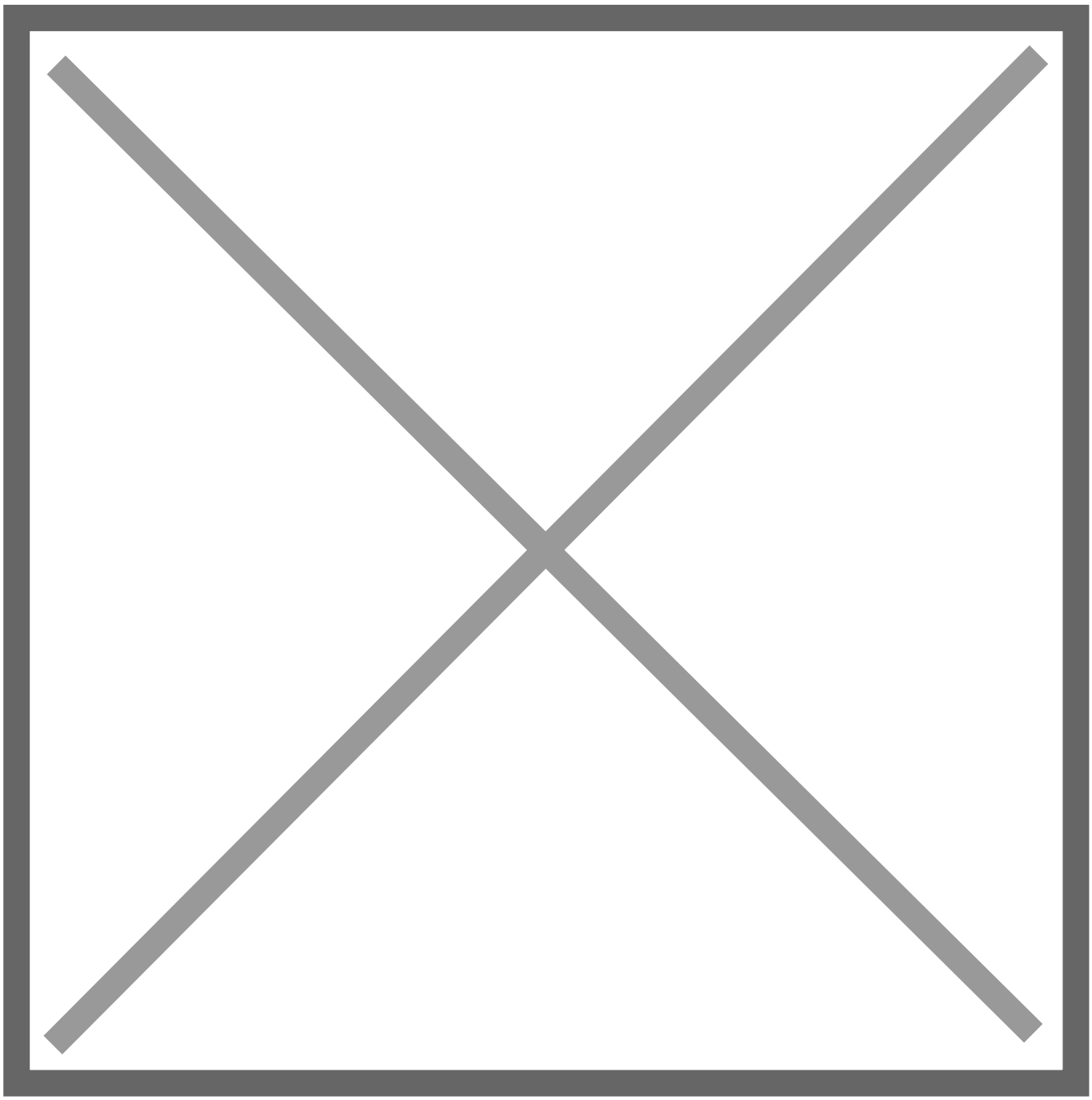
Рассмотрим события из вкладки Security, в котором могут быть сработки сигнатур для выявления различных аномалий, где мы как раз видим алерты сигнатуры TCP Syn Port Scan с IP-адреса 192.168.1.8 нацеленного на IP-адрес 192.168.1.2, как раз в 17:55.



Если открыть алерт со сканированием, то можно увидеть входящие в него события, где уже видно какие порты были просканированы.



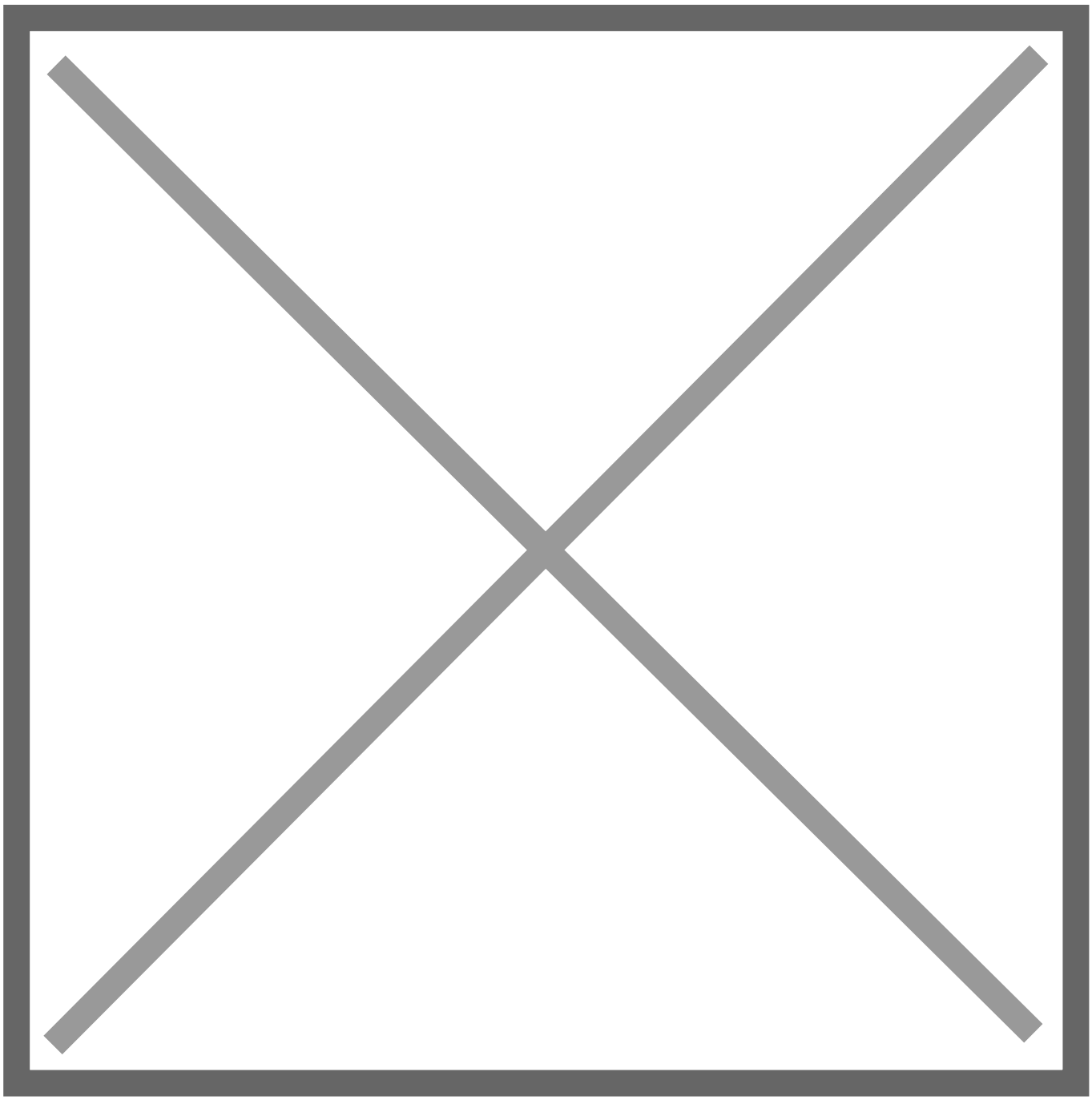
Если дополнительно проанализировать на атакуемом хосте с IP-адресом 192.168.1.2 журналирование событий Sysmon с EventID = 3 (Network Connection), то можно увидеть большое количество сетевых соединений с различными портами, что также констатирует об активности сканирования портов.



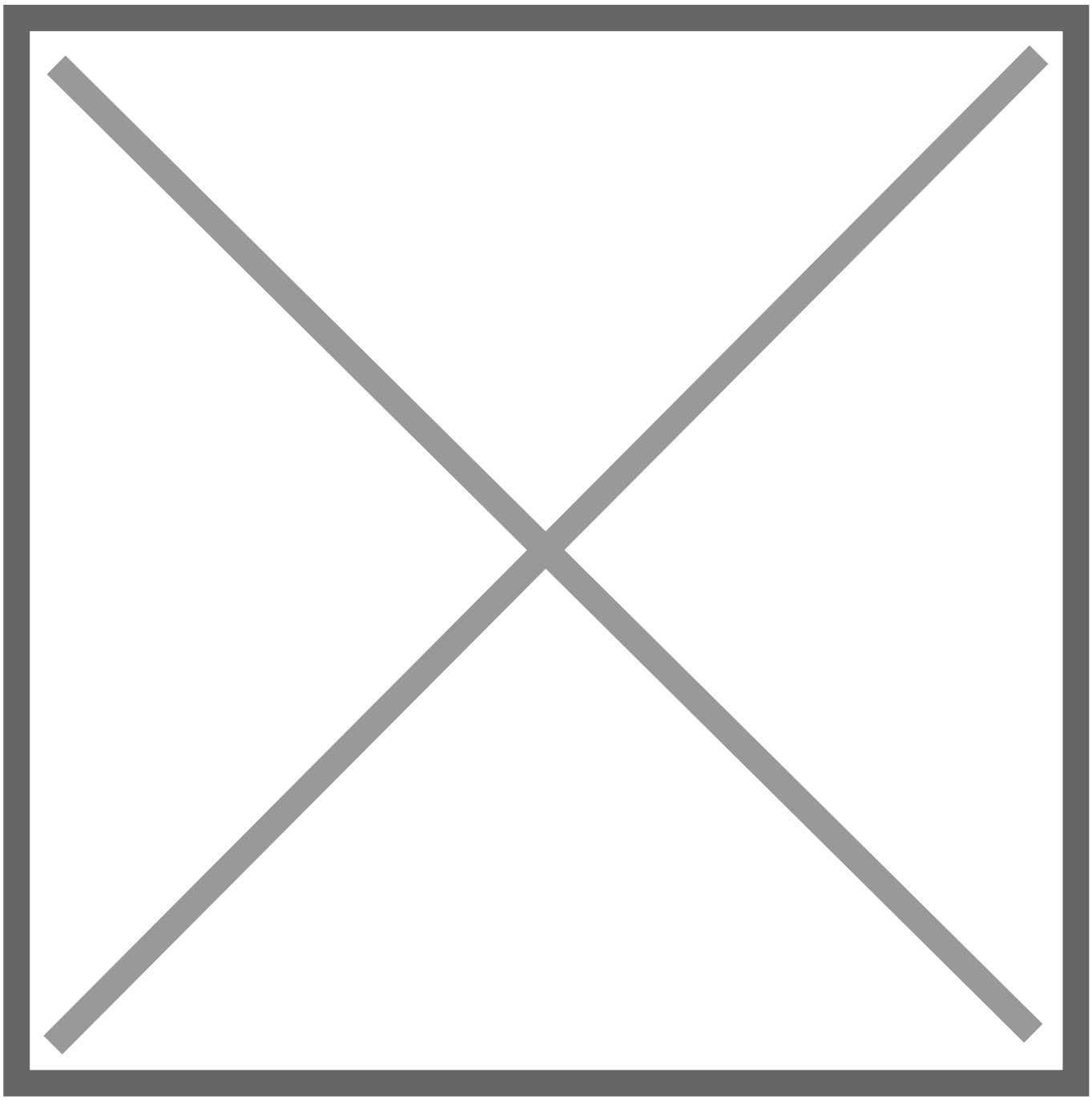
Отсюда можно сделать вывод, что для выявления сканирований различных портов на одном хосте, необходимо отслеживать пиковые исходящие активности от одного хоста инициатора и анализировать их.

В качестве альтернативного решения для сбора, нормализации, визуализации и анализа NetFlow трафика может быть использован netflow модуль для logstash, который тоже поддерживает NetFlow 5 и 9 версии.

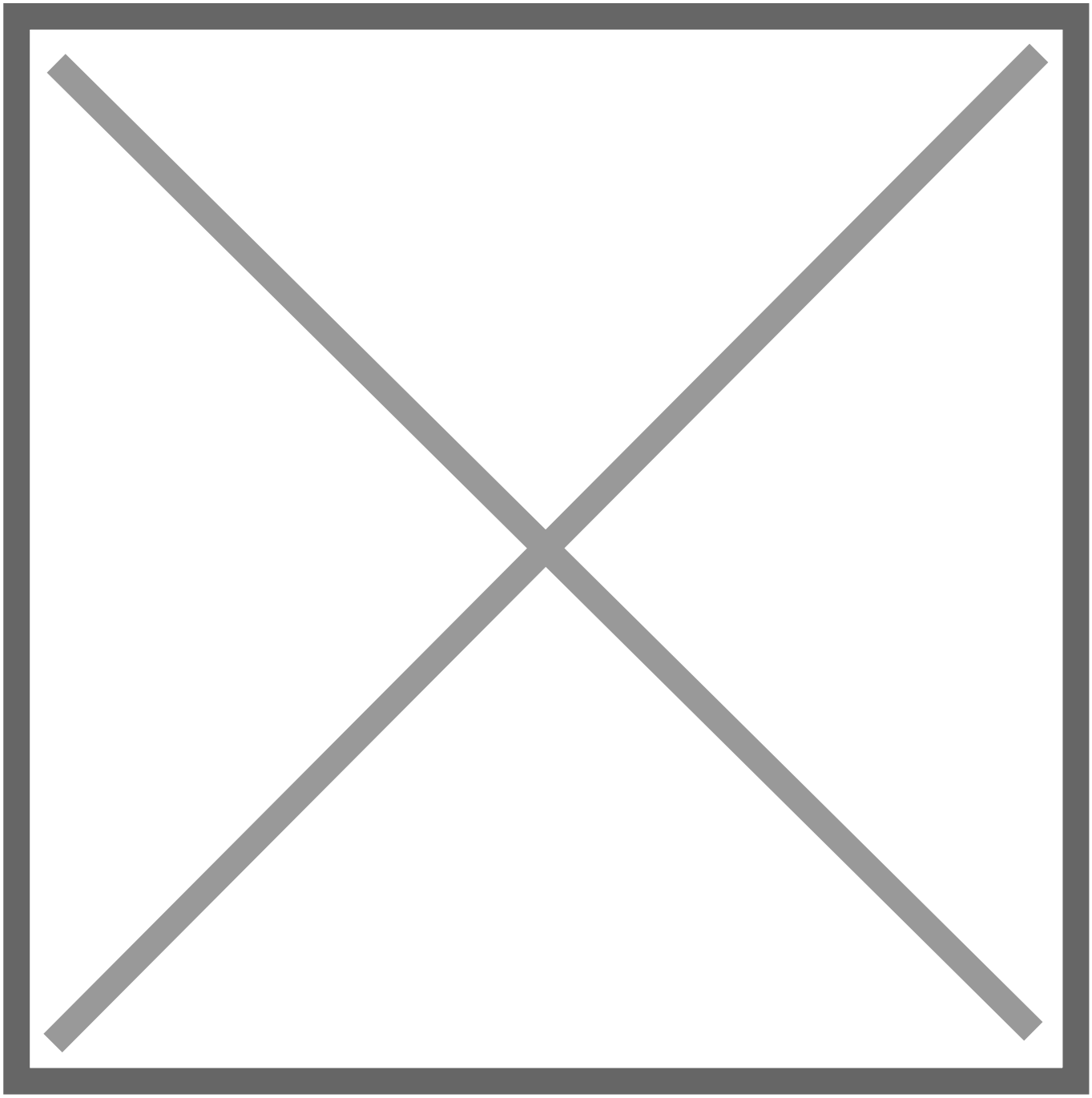
На панели «Overview» размещена сводка основных данных о сетевом трафике, возможно настроить фильтры.



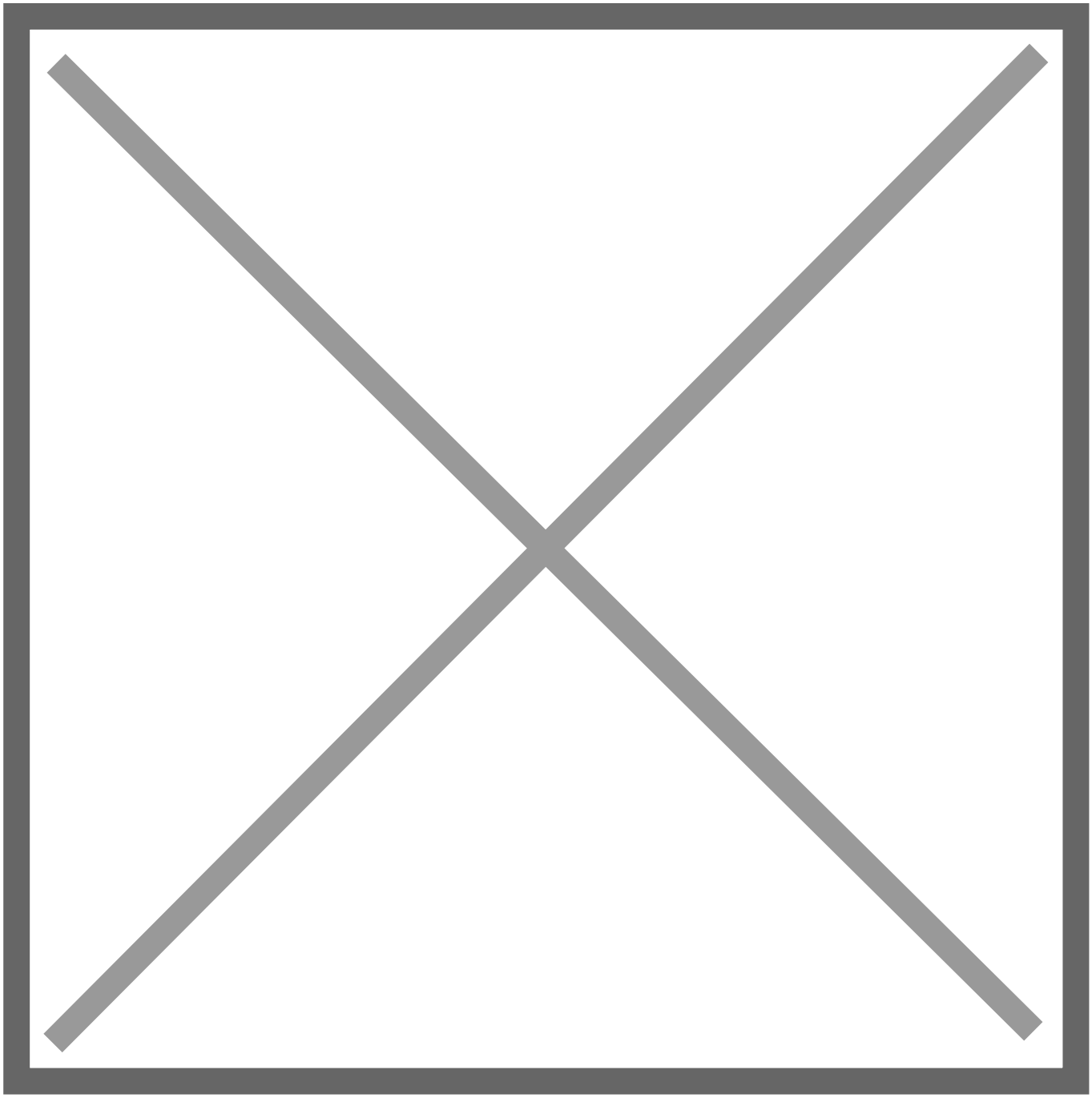
На панели «Conversation Partners» IP-адреса отправителя, получателя и объем передаваемого трафика.



На панели **«Traffic Analysis»** можно более детально проанализировать объемы передаваемого трафика за конкретный временной период.



Затем можете перейти на панель **«Geo Location»**, где визуально посмотрите тепловую карту с потоком сетевого трафика.



Обнаружение подозрительной активности в событиях сетевых устройств

Сетевое оборудование и межсетевые экраны журналируют события сетевых соединений, действия в системе и изменения в собственных настройках в стандарте syslog. Но фиксация кажового события сетевого соединения часто невозможна на высоконагруженных сетях из-за большого объема событий, поэтому менее затратным, в плане дисковой подсистемы, является мониторинг статистики сетевых соединений NetFlow.

События сетевого устройства Cisco

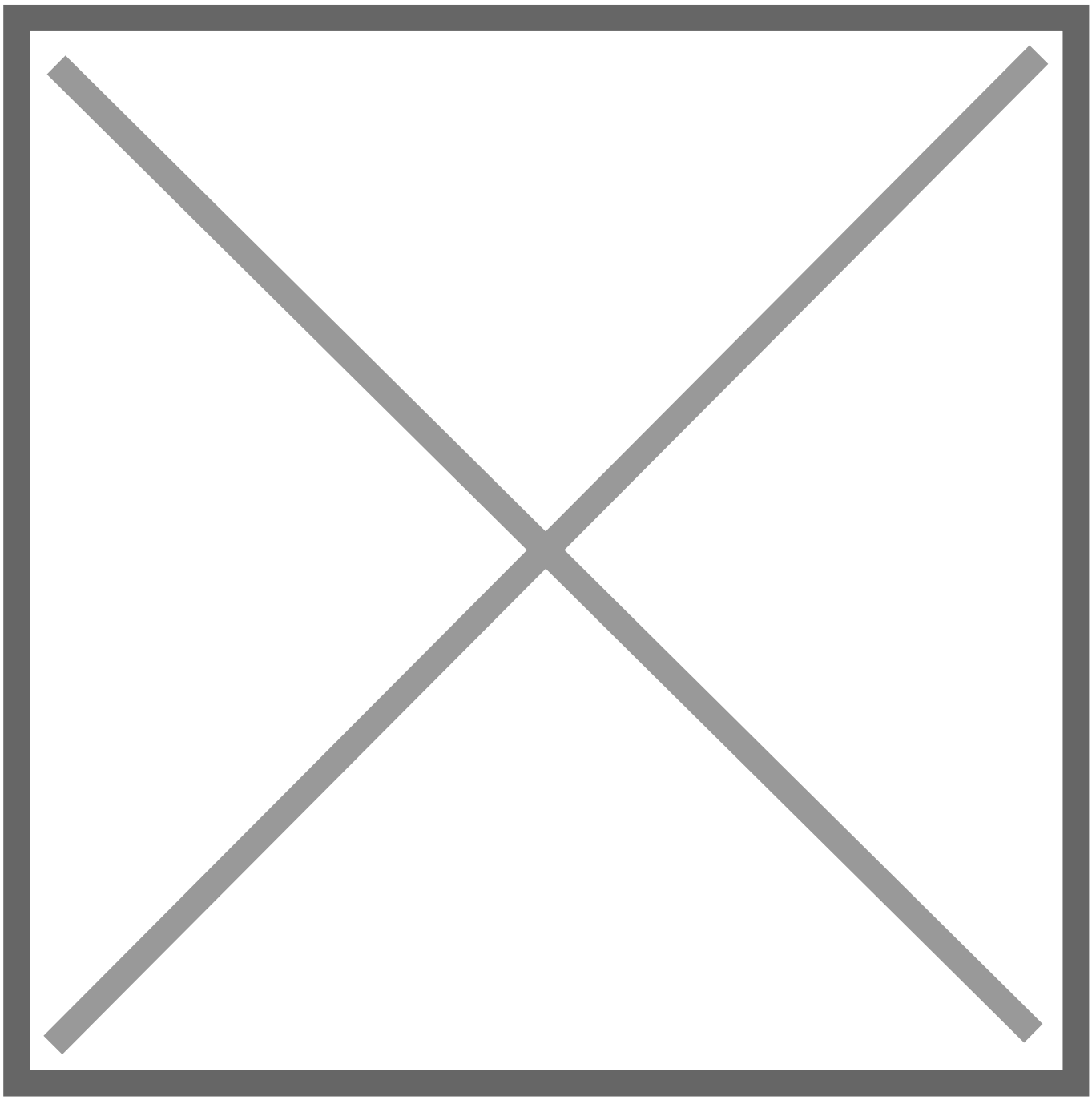
Рассмотрим примеры событий с Cisco ASA, у которого подробная документация по описанию событий. В рамках начального курса не будем углубляться в детали настройки системы

журналирования. При необходимости подробно можно ознакомиться с конфигурированием аудита на сайте вендора.

```
May  5 19:02:25 dev01: %ASA-3-113021: Attempted console login failed. User adm did NOT have appropriate Admin Rights.
May  5 19:02:25 dev01: %ASA-6-716039: Authentication: rejected, group = lab user = adm , Session Type: admin
May  5 19:02:25 dev01: %ASA-6-716039: Authentication: rejected, group = lab user = ivan , Session Type: WebVPN
May  5 19:02:25 dev01: %ASA-6-716039: Group <lab> User <adm> IP <172.31.98.44> Authentication: rejected, Session Type: Admin.
May  5 19:02:25 dev01: %ASA-6-716039: Group <lab> User <ivan> IP <172.31.98.44> Authentication: rejected, Session Type: WebVPN.
```

Если обратить внимание, то события Cisco имеют структуру, где:

- `May  5 19:02:25` - месяц, день и время регистрации события;
- `dev01` - наименование устройства, на котором зарегистрировано событие;
- `%ASA-3-113021` - идентификатор события, описание события в документации вендора;
- `Attempted console login failed. User eve did NOT have appropriate Admin Rights -` описание/тело события.



Соответственно первое событие

```
May  5 19:02:25 dev01: %ASA-3-113021: Attempted console login failed. User adm did NOT have
appropriate Admin Rights
```

фиксирует неудачную попытку аутентификации к консоли администрирования сетевого оборудования Cisco под учетной записью **adm** с правами администратора.

В следующем событии

```
May  5 19:02:25 dev01: %ASA-6-716039: Group <lab> User <adm> IP <172.31.98.44> Authentication:
rejected, Session Type: Admin
```


зафиксирована информация, с какого IP-адреса(172.31.98.44) была неудачная попытка аутентификации к консоли сетевого оборудования под учетной записью **adm,** где так же указан тип сессии Admin, который позволит нам отличить от других типов соединений, таких как подключений к VPN, у которых будет тип сессии WebVPN `Session Type: WebVPN.`

Таким образом можно отслеживать несанкционированные попытки подключения, подбора паролей консоли управления сетевого оборудования для учетных записей сетевых устройств.

Следующий набор событий для примера:

```
May  5 19:03:27 dev01: %ASA-7-111009: User 'pavel' executed cmd: show access-list
fw211111_access_out brief
May  5 19:02:26 dev01: %ASA-7-111009: User 'pavel' executed cmd: show access-list aaa_out
brief
Apr 27 02:03:03 dev01: %ASA-5-111004: console end configuration: OK
Apr 27 02:03:03 dev01: %ASA-5-111010: User 'enable_15', running 'CLI' from IP 10.10.0.87,
executed 'clear'
Apr 27 02:03:03 dev01: %ASA-5-502103: User priv level changed: Uname: enable_15 From: 1 To: 15
```


Событие с идентификатором `%ASA-7-111009` фиксирует любые команды без изменения конфигурации, выполненные в консоли управления сетевым оборудованием Cisco. В данном случае зафиксировано выполнение команды по выводу листа с разрешениями `show access-list fw211111_access_out brief.`

Событие с идентификатором `%ASA-5-111010` фиксирует изменение конфигурации оборудования под учетной записью `enable_15`, при это журналируется IP-адрес откуда был запущена консоль управления `running 'CLI' from IP 10.10.0.87`, в данном случае выполнена команда `clear`.

Событие с идентификатором `%ASA-5-502103` фиксирует изменение уровня привилегий у учетной записи `enable_15` от 1 до 15 `From: 1 To: 15.` У оборудования Cisco всего используется 15 уровней привилегий. Уровень привилегий 1 (**privilege 1**). Соответствует пользовательскому режиму (т.е. в качестве приглашения в командной строке **switch>**). Команды из привилегированного режима недоступны. Уровень привилегий 15 (**privilege 15**). Привилегированный режим, где доступны все команды (приглашение в командной строке **switch#**).

Фильтрация

Фильтрация исходящего трафика необходима прежде всего для предотвращения утечек данных и контроля использования внешних ресурсов. Основные методы:

- **Stateful Packet Inspection (SPI).** Этот метод фильтрации анализирует сетевые сессии, а не просто отдельные пакеты данных. В отличие от более простой фильтрации пакетов, которая решает принимать или отклонять пакеты на основе их содержимого (например, IP-адреса и порта), SPI учитывает контекст и состояние каждого соединения.
- **Application Layer Filtering.** Фильтрация на уровне приложения позволяет анализировать трафик, ориентируясь на конкретные приложения или службы. Это может включать в себя блокировку определенных протоколов или служб, таких как FTP, HTTP, или использование специализированных систем контроля контента.
- **URL Filtering.** Фильтрация URL предотвращает доступ к определенным веб-сайтам или категориям веб-сайтов. Это может быть полезным для управления производительностью, обеспечения безопасности и соблюдения корпоративных политик.
- **Data Loss Prevention (DLP).** Технологии DLP помогают предотвращать утечку конфиденциальных данных, блокируя передачу конфиденциальной информации в исходящем трафике.

Инструменты и технологии:

- **Брандмауэры и межсетевые экраны (Firewalls).** Они обычно предоставляют основные средства фильтрации исходящего трафика.
- **Прокси-сервера.** Прокси-серверы могут использоваться для контроля доступа, фильтрации содержимого и анализа трафика.
- **Системы предотвращения утечки данных (DLP systems).** DLP-системы предназначены для обнаружения и блокировки утечек конфиденциальной информации.
- **Системы Content Filtering и URL Filtering.** Используются для фильтрации по категориям, блокировке веб-сайтов и контроля за тем, какие типы контента могут быть доступны через сеть.
- Эффективная фильтрация исходящего трафика важна для обеспечения безопасности и эффективности работы сети, а также для соблюдения корпоративных политик и нормативных требований.

Фильтрация входящего трафика необходима для защиты от атак, направленных снаружи. Основные методы фильтрации входящего трафика:

- **Stateful Packet Inspection (SPI).** Аналогично с фильтрацией исходящего трафика, выполняется проверка сессий на предмет аномальных взаимодействий.
- **Блокировка по IP-адресам и портам.** Администраторы могут определить правила фильтрации для блокировки трафика с определенных IP-адресов или портов. Это может быть использовано для предотвращения атак, связанных с известными

вредоносными IP-адресами или портами.
- **Антивирусная фильтрация.** Входящий трафик проверяется на предмет вредоносной активности.
- Content Filtering: В контексте фильтрации входящего трафика фильтрация контента относится к защите от фишинговых почтовых рассылок
- **Intrusion Detection Systems (IDS) и Intrusion Prevention Systems (IPS).** Эти системы обнаруживают и предотвращают вторжения, анализируя входящий трафик по пакетам на предмет аномалий и подозрительного поведения.

Инструменты и технологии:

- **Межсетевые экраны, в том числе WAF.** Позволяют фильтровать входящий сетевой трафик по определенным критериям (адреса, порты, заголовки HTTP и т.д.)
- **IDS/IPS.** С помощью сигнатурного анализа позволяют выявлять и/или блокировать подозрительный трафик
- **Content Filtering Systems.** Блокируют взаимодействия с потенциально опасным контентом, в том числе защищают от фишинговых рассылок

Подходы к анализу трафика

Сначала необходимо определить, что именно мы будем искать. Например, мы знаем, что злоумышленник закрепился на конкретном хосте, тогда нас будет интересовать входящий и исходящий трафик относительно этого хоста. Настроив фильтры при захвате трафика или при поиске в уже существующих дампах, мы можем приступать к анализу.

На что можно опираться при анализе трафика:

- Зашифрован трафик или нет? Должен ли он быть таковым?
- Пытались ли потенциальные злоумышленники получить доступ к ресурсам, доступ к которым для них ограничен?
- Взаимодействуют ли между собой хосты, которые обычно такого не делают?
- Возникают ли ошибки, как отвечают на запросы хосты?
- Есть ли что-то, что выбивается из общей статистики в трафике?

На этом этапе могут пригодиться другие инструменты вроде IDS и IPS. Опираясь на сигнатурный анализ, можно выявить не легитимный трафик и исходя из полученной информации развивать поиск.

Анализ сетевого трафика является динамическим процессом, способным изменяться в зависимости от доступных инструментов и "прозрачности" нашей сети (передается ли трафик в зашифрованном или в открытом виде).

Здесь важно уметь делать разделение данных на доступные для понимания фрагменты, изучение их на предмет отличий от обычного трафика и на потенциально вредоносный трафик вроде неавторизованных подключений через интернет с помощью RDP, SSH или Telnet. Выполняя анализ, мы также пытаемся определить существующие в трафике тренды и понять, насколько они соотносятся с обычным трафиком.

Практические советы по анализу трафика:

Поиск лучше всего начинать со стандартных протоколов, переходя к более редким постепенно по мере анализа. Большая часть атак происходит из интернета и поэтому требует получения доступа во внутреннюю сеть. Это означает появление трафика и создание касающихся его логов. HTTP/S, FTP, E-mail и обычный TCP и UDP трафик будет наиболее распространенным входящим внешним трафиком. Начинайте с него, фильтруя все, что не касается расследования. После этого проверьте все стандартные протоколы, обеспечивающие связь между сетями - SSH, RDP или Telnet. Когда вы ищете подобные аномалии, учитывайте политику безопасности сети. Допускает ли политика безопасности нашей организации сессии RDP, инициируемые извне? А использование Telnet?

Ищите паттерны. Один или несколько хостов регулярно сверяются с чем-то в интернете в одно и то же время? Это обычное поведение при взаимодействии с CnC серверами.

Обращайте внимание на уникальные события. Отличающаяся от обычных приложений строка Юзер-Агента или подключающиеся к внешнему серверу хосты также требуют внимания. Подозрение вызывает также привязанный только единожды или дважды порт - он может быть открытым для обратного CnC-трафика, для нестандартных действий с чьей-то стороны или же для аномально функционирующего приложения.

Не бойтесь просить помощи. Во-первых, после длительного просмотра дампов и логов можно упустить из вида важные детали, которые может заметить "свежий взгляд". Во-вторых, при разборе инцидента часто необходимо использовать другие компетенции помимо анализа сетевого трафика - например, просматривать логи DNS-сервера и контроллера домена.

Revision #2
Created 25 October 2025 11:57:51 by Admin
Updated 27 October 2025 15:47:48 by Admin