Составитель Андрей Областов

Форензика

Методическое пособие для студентов

Андрей Александрович Обласов

Форензика. Методическое пособие для студентов

http://www.litres.ru/pages/biblio_book/?art=56557213
ISBN 9785005110688

Аннотация

Форензика, простыми словами, – это сбор цифровых доказательств. Что можно понять под словосочетанием «цифровые доказательства». Компьютерная криминалистика (Форензика).Под цифровыми доказательствами подразумевается наличие следов преступления на любых носителях информации: жесткие диски, смартфоны, планшеты, флешки, любые предметы, где может храниться информация в электронном виде. Из самых не обычных – это могут быть утюги, холодильники и прочая умная бытовая техника:)

Содержание

Форензика
Методическое пособие для студентов

Составитель Андрей Александрович Обласов

ISBN 978-5-0051-1068-8
Создано в интеллектуальной издательской системе Ridero

ВВЕДЕНИЕ

Форензика простыми словами, это сбор цифровых доказательств. Что можно понять под словосочетанием «цифровые доказательства». Под цифровыми доказательствами подразумевается наличие следов преступления на любых носителях информации, жесткие диски, смартфоны, планшеты, флешки, любые предметы где может храниться информация в электронном виде. Из самых не обычных это могут быть утюги, холодильники и прочая умная бытовая техника))

Существует несколько видов криминалистических экспертиз:

Аппаратно-компьютерная экспертиза;

Программно-компьютерная экспертиза;

Компьютерно-сетевая экспертиза;

Информационно-компьютерная экспертиза.

Перед специалистом в основном ставятся вопросы:

о наличии на объектах информации, которая относится к делу (в том числе, в неявном, удалённом, скрытом или зашифрованном виде);

о возможности использования исследуемых объектов для определённых целей (например, для доступа в сеть);

о действиях, совершённых с использованием объектов;

о свойствах программ, в частности, о принадлежности их

к вредоносным;

об идентификации найденных электронных документов, софта, пользователей компьютера.

Еще есть понятие как экспресс анализ, то есть на месте быстро все проверяется и делается заключение. Например, вы сидите дома, взламываете сайт какого нибудь департамента, и тут в дверь позвонили, ты пошел открывать не выключив ноутбук, зашли полицейские, или люди из иных гос. структур, посмотрели ваш компьютер, и все, доказательство на лицо. По средствам экспресс анализа быстро выявят ваше причастие к взлому и заберут в другое место, менее приятнее чем ваш дом.

1 Общие сведения о логе веб-сервера

1.1 Как работает web server log file?

Когда пользователь вводит URL-адрес в браузер, тот сначала разбивает его на три компонента. Например:

https://your_site_address.com/example.html

В данном случае браузер понимает, что https – это протокол, your_site_address.com – название сервера, а example.html – имя файла.

Название сервера преобразуется в IP-адрес через сервер доменных имен. Затем HTTP-запрос GET отправляется на веб-сервер через соответствующий протокол для запрашиваемой страницы или файла, при этом HTML возвращается в браузер, а затем интерпретируется для форматирования видимой страницы на экране. Каждый из этих запросов записывается в log file веб-сервера.

Проще говоря, процесс выглядит так: посетитель совершает переход по странице, браузер передает его запрос серверу, на котором расположен веб-сайт. Сервер выдает запрошенную пользователем страницу в ответ. И после этого фиксирует все происходящее в log-файле.

Все, что вам нужно, чтобы проанализировать сканирование сайта поисковой системой, – экспортировать данные и отфильтровать запросы, сделанные роботом, например, Googlebot. С помощью браузера и диапазона IP это сделать удобнее.

Сам лог-файл представляет собой сырую информацию, сплошной текст. Но правильная обработка и анализ дают неограниченный источник информации.

1.2 Содержание и структура лог-файла

Структура журнала в конечном счете зависит от типа используемого сервера и конфигураций. Например, анализ логов Apache будет отличаться от анализа логов Nginx. Но есть несколько общих атрибутов, которые почти всегда содержатся в файле:

- IP-адрес запроса;
- дата и время;
- география;
- метод GET / POST;
- запрос к URL;
- код состояния HTTP;
- браузер.

Пример записи, включая приведенные выше данные:

111.11.111.111 – - [12 / Oct / 2018: 01: 02: 03 -0100] « GET / resources / whitepapers / retail-whitepaper / HTTP / 1.1 « 200“ -« „Opera / 1.0 (совместимый; Googlebot / 2.1; + http:// www.google.com/bot.html)

Дополнительные атрибуты, которые иногда можно увидеть, включают:

- имя хоста;
- запрос / клиентский IP-адрес;
- загруженные байты;

• затраченное время.

Экспорт лог-файла на WordPress

Чтобы подобный файл появился и для сайта на платформе WordPress, необходимо включить функцию логирования. Для этого найдите в корневой папке сайта файл с названием wp-config.php. Скачайте файл на компьютер, чтобы получить доступ к редактированию.

Далее найдите строку: *«That's all, stop editing! Happy blogging».* Перед ней добавьте новую строку кода:

define («WP_DEBUG», true);

Это переведет сайт в режим отладки, что включит отображение уведомлений об ошибках.

Теперь запустите запись ошибок в log-файл. Для этого сразу под предыдущей строкой кода добавьте новую:

define («WP_DEBUG_LOG», true);

Чтобы зайти в log-файл сайта на WordPress, нужно перейти в FTP или менеджер файлов. Далее в общей папке сайта откройте папку wp-content, в ней найдите файл с именем debug.

1.3 Анализаторы файлов журналов

По мере того, как все больше и больше компаний переходят в облако, аналитики журнала, анализ логов и инструменты управления журналами становятся все более востребованными.

Некоторые проверяют лог-файлы вручную. Веб-мастера экспортируют файл и анализируют его в программе Excel. В таком случае понадобится только сортировка и несколько формул, но данный подход – устаревший, и здесь рассматриваться не будет.

Использование специальных инструментов для анализа файлов журналов может облегчить обработку больших объемов информации.

Некоторые веб-мастера устанавливают анализатор логов на сам сервер. Способ удобен для проектов, расположенных на собственных серверах, – тогда логи будут сохранены неограниченное количество времени. А проекты, которые находятся на стороннем хостинге, будут поставлены в рамки – хранение максимум 1 месяц. Поэтому возникает необходимость производить ротацию и архивирование.

Screaming Frog Log File Analyzer

Для примера проведем анализ лог файлов через Screaming Frog Log File Analyzer.

Инструмент предоставляет доступ к бесплатной версии, ограничивая журнал событий одной тысячей строк. Для некрупного проекта этого объема хватит.

Скачайте и установите программу на компьютер, далее выгрузите лог-файлы или составьте список всех URL-адресов, которые присутствуют на веб-площадке. Экспорт файла описан выше.

Анализ логов онлайн покажет в таблице коды ответа страниц, даты последних переходов, контент, количество поисковых ботов и прочее.

Остальные анализаторы рассмотрим более кратко.

GoAccess предназначен для быстрого анализа данных. Его основная идея – быстро посмотреть логи сервера и проанализировать их в режиме реального времени без необходимости использования вашего браузера.

Splunk позволяет вам обрабатывать бесплатно до 500 МБ данных в день. Это отличный способ собирать, хранить, искать, сравнивать, анализировать журналы сайта.

Logmatic.io – инструмент анализа журналов, разработанный специально для улучшения работы программного обеспечения. Акцент делается на программные данные, куда входят и журналы. На данный момент инструмент платный.

Logstash – бесплатный инструмент с открытым исходным кодом для управления событиями и журналами. Его можно использовать для сбора журналов, их хранения и анализа.

Большинство ОС для web-серверов предоставляют воз-

можность указать конкретные привилегии доступа для файлов, устройств и других вычислительных ресурсов на данном хосте. Следует понимать, что любая информация, к которой может быть осуществлен доступ из каталогов, принадлежащих web-серверу потенциально может стать доступной всем пользователям, имеющим доступ к публичному web-сайту. Обычно ПО web-сервера обеспечивает дополнительное управление доступом к файлам, устройствам и ресурсам. В случае, если разрешения доступа к ресурсам могут быть установлены как на уровне ОС, так и на уровне приложения web-сервера, важно, чтобы они были идентичны друг другу, в противном случае возможна ситуация, когда пользователям предоставлен либо очень большой, либо очень маленький доступ. Web-администраторы должны рассмотреть, какое конфигурирование доступа к хранимой в публичном web-сервере информации является наилучшим со следующих точек зрения:

• ограничение доступа ПО web-сервера к подмножеству вычислительных ресурсов средствами управления доступом ОС;

• ограничение доступа пользователей посредством дополнительного управления доступом, обеспечиваемым web-сервером, когда требуются более детальные уровни управления доступом.

Соответствующая установка управления доступом может помочь предотвратить раскрытие чувствительной информа-

ции, которая не предназначена для публичного распространения. Дополнительно управление доступом может быть использовано для ограничения использования ресурсов в случае DoS-атаки на публичный web-сайт.

Обычно файлами, доступ к которым должен контролироваться, являются следующие:

- ПО и конфигурационные файлы приложения.
- Файлы, непосредственно использующиеся механизмами безопасности: файлы хэшей паролей и другие файлы, используемые при аутентификации; файлы, которые содержат авторизационную информацию, используемую при управлении доступом; криптографический материал ключа, используемый в сервисах конфиденциальности, целостности и невозможности отказа.
- Логи сервера и файлы системного аудита.
- Системное ПО и его конфигурационные файлы.

1.4 Разграничение доступа для ПО web-сервера

Первый шаг в конфигурировании управления доступа состоит в гарантировании того, что web-сервер выполняется только от имени пользователя и группы, которые специально созданы для этого и имеют очень ограниченные права доступа. Таким образом, должны быть введены специальные идентификаторы пользователя и группы, используемые исключительно ПО web-сервера. Новый пользователь и новая группа должны быть уникальными и независимыми от всех остальных пользователей и групп. Это необходимо для реализации управления доступом, описанного далее. Хотя сервер может начинать выполняться как root (Unix) или system/administrator (Windows NT/2000/XP) для привязки к TCP-порту 80 и/или 443 (используемому для предоставления HTTP и HTTPS-сервисов соответственно), не следует допускать, чтобы сервер продолжал выполняться на данном уровне доступа.

Дополнительно следует использовать возможности ОС для ограничения доступа к файлам, доступным процессам web-сервера. Эти процессы должны иметь доступ только по чтению к тем файлам, которые необходимы для выполнения сервиса, и не должны иметь доступа к остальным файлам, таким как файлы лога сервера. Следует использовать

управление доступом на уровне ОС для обеспечения следующего:

• процессы web-сервера должны быть сконфигурированы для выполнения от имени пользователя с очень ограниченным множеством привилегий (т.е. не выполняться как root, администратор или эквивалентные пользователи);

• к файлам содержимого web-сайтов процессы web-сервера должны иметь доступ по чтению, но не по записи;

• процессы web-сервера не должны иметь возможность записи в директории, в которых хранится публичное содержимое web-сайтов;

• только процессы, авторизованные как администратор web-сервера, могут писать в файлы web-содержимого;

• приложение web-сервера может писать в файлы логов web-сервера, но лог-файлы не могут читаться приложением web-сервера. Только процессы уровня root/system/administrator могут читать лог-файлы web-сервера;

• временные файлы, создаваемые приложением web-сервера, например, те, которые возникают при формировании динамических web-страниц, должны быть расположены в специальной и соответствующим образом защищенной поддиректории;

• доступ к любым временным файлам, созданным приложением web-сервера, ограничен процессами, которые создали эти файлы.

Также необходимо гарантировать, что приложение web-

сервера не может хранить файлы вне специального подкаталога, выделенного для публичного web-содержимого. Это может быть задано посредством конфигурации в ПО сервера или может контролироваться ОС. Следует гарантировать, что к директориям и файлам вне специального поддерева директорий не может быть обращений, даже если пользователи знают имена этих файлов.

Для уменьшения воздействия основных типов DoS-атак нужно сконфигурировать web-сервер с ограниченным количеством ресурсов ОС, которые он может использовать. Чаще всего необходимо совершить следующие действия:

• инсталлировать содержимое web на отдельном жестком диске или логическом разделе от ОС и web-приложения;

• если допустимы загрузки (uploads) на web-сервер, установить ограничение на объем дискового пространства, которое выделяется для этой цели;

• если допустимы загрузки (uploads) на web-сервер, эти файлы не должны быть сразу же читаемы web-сервером и, тем самым, видимы пользователям по протоколу HTTP. Они должны быть читаемы web-сервером только после некоторого автоматизированного или ручного процесса просмотра. Это предотвращает от использования web-сервера для передачи пиратского ПО, инструментальных средств атак, порнографии и т.п.;

• гарантировать, что лог-файлы хранятся в соответствующем месте, в котором они не смогут исчерпать ресурсы фай-

ловой системы.

Эти действия в некоторой степени защитят от атак, которые попытаются заполнить файловую систему информацией, что может вызвать крах системы. Они могут также защитить против атак, которые пытаются заполнить RAM память ненужными процессами для замедления или краха системы, тем самым ограничив доступность сервиса. Информация в логах, созданных ОС, может помочь распознать такие атаки.

Дополнительно часто бывает необходимо сконфигурировать таймауты и другие способы управления для дальнейшего уменьшения влияния основных DoS-атак. Один из типов DoS-атаки состоит в том, чтобы одновременно устанавливать сетевые соединения сверх максимально допустимого, чтобы никакой новый законный пользователь не мог получить доступ. Когда таймауты установлены на сетевые соединения (время, после которого неактивное соединение сбрасывается) в минимально допустимое ограничение, существующие соединения будут завершаться по таймауту так быстро, как только возможно, создавая возможность устанавливать новые соединения законным пользователям. Данная мера только смягчает DoS-атаку, но не уничтожает ее.

Если максимальное число открытых соединений (или соединений, которые являются полуоткрытыми, – это означает, что первая часть TCP-рукопожатия завершилась успешно) установить в наименьшее число, атакующий может лег-

ко израсходовать доступные соединения ложными запросами (часто называемыми SYN flood). Установка в максимум данного числа может смягчить эффект такой атаки, но ценой расходования дополнительных ресурсов. Заметим, что это является проблемой только тех web-серверов, которые не защищены firewall'ом, останавливающим SYN flood атаки. Большинство современных firewall'ов защищают web-сервер от SYN flood атаки, прерывая ее прежде, чем она достигнет web-сервера.

1.5 Управление доступом к директории содержимого web-сервера

Не следует использовать ссылки, aliases или shortcuts из дерева директории содержимого web на директории или файлы, расположенные где-то еще на хосте или сетевой файловой системе. Лучше всего запретить возможность ПО web-сервера следовать по ссылкам и aliases. Как указывалось раньше, лог-файлы и конфигурационные файлы web-сервера должны размещаться вне дерева директории, которое определено для публичного содержимого web.

Для ограничения доступа к дереву директории содержимого web требуется выполнить следующие шаги:

• выделить отдельный жесткий диск или логический раздел для web-содержимого и использовать поддиректории на данном жестком диске или логическом разделе исключительно для файлов содержимого web-сервера, включая графику, но исключая скрипты и другие программы;

• определить отдельную директорию исключительно для всех внешних по отношению к ПО web-сервера скриптов или программ, выполняющихся как часть содержимого web (например, CGI, ASP, PHP);

• запретить выполнение скриптов, которые не находят-

ся под исключительным контролем административных аккаунтов. Данное действие выполняется созданием доступа и управлением им к отдельной директории, предназначенной для хранения авторизованных скриптов;

• запретить использование жестких и символических ссылок;

• определить матрицу доступа к содержимому web. Определить, какие папки и файлы внутри содержимого web-сервера имеют ограничения и какие являются доступными (и кому).

Большинство производителей ПО web-сервера предоставляют директивы или команды, которые позволяют web-администратору ограничить доступ пользователя к файлам содержимого web-сервера.

Например, ПО web-сервера Apache предоставляет директиву Limit, которая позволяет web-администратору указать, какие возможности доступа (такие как New, Delete, Connect, Head и Get) связаны с каждым файлом web-содержимого. Директива Require в Apache позволяет web-администратору ограничивать доступ к содержимому аутентифицированными пользователями или группами.

Многие директивы или команды могут быть перекрыты на уровне директории. Но при этом следует помнить, что удобство, состоящее в возможности делать локальные исключения из глобальной политики, может привести к появлению дыры в безопасности, появившейся в отдельной под-

директории, которая контролируется другим пользователем. Web-администратор должен запретить для поддиректорий возможность перекрывать директивы безопасности верхнего уровня, если это не является абсолютно необходимым.

В большинстве случаев подобные директивы web-сервера должны быть запрещены. HTTP определяет, что URL, заканчивающийся символом слеша, должен трактоваться как запрос на перечисление файлов в указанной директории. Web-сервер должен запрещать отвечать на такие запросы, даже если возможно публичное чтение всех файлов директории. Такие запросы часто указывают на попытку разместить информацию способами, отличными от тех, которые допускает web-администратор. Пользователи могут пытаться это делать, если у них возникают трудности в навигации по сайту или если ссылки в web-страницах обрываются. Нарушитель может пытаться это сделать, чтобы разместить информацию, скрытую интерфейсом сайта. Web-администраторы должны исследовать запросы данного типа, найденные в лог-файлах web-сервера.

На большинстве современных хостингов кроме FTP доступа к файловой системе предоставляется также SSH доступ (по-умолчанию или по запросу в тех поддержку). Умение веб-мастера работать с файлами сайта в терминале (в режиме командной строки) по SSH экономит ему массу времени. Операция, которая может занимать десятки минут по FTP, делается через командную строку за пару секунд.

Кроме того, есть много операций, которые можно сделать только по SSH в режиме командной строки.

Веб-мастеру не обязательно осваивать весь инструментарий операционной системы Unix, для начала достаточно познакомиться с базовыми командами, а к ним добавить несколько полезных трюков при работе с командной строкой по SSH, чтобы быстро искать файлы, изменять их атрибуты, копировать, удалять и выполнять операции с текстовыми данными.

Я пропущу описание протокола и процесса подключения к аккаунту хостинга по SSH, в сети можно найти множество видео-уроков и статей по данной теме, скажу лишь что для подключения вам потребуется программа Putty (ОС Windows) / Терминал (Mac OS X) или аналогичные, и доступы к хостингу по SSH: хост, порт, логин и пароль (часто имя и пароль они совпадают с доступом в cPanel, ISPManager или аккаунтом панели управления хостингом).

Итак, что полезного можно делать в командной строке? Можно быстро выполнять поиск подстроки в текстовом файле, сортировку, фильтрацию текстовых данных. Например, для анализа журналов (логов) веб-сервера, чтобы выявить подозрительные запросы к сайту или понять, как взломали сайт.

Предположим, вы заметили подозрительную активность на сайте (стал медленно открываться, пропали доступы в админ-панель, с сайта рассылают спам и т.п.). Первое, что

в этом случае нужно выполнить – это проверить файлы сайта на вредоносный код специализированными сканерами. Но пока сайт сканируется, можно провести экспресс-анализ логов веб-сервера с помощью команд find/grep, чтобы опеределить, не было ли обращений к каким-то подозрительным скриптам, попыток брутфорса (подбора пароля) или вызовов хакерских скриптов. Как это сделать? Об этом ниже.

Чтобы анализировать журналы (логи) веб-сервера нужно, чтобы эти логи были включены и доступны в пользовательской директории. Если по-умолчанию они отключены, необходимо их включить в панели управления хостингом и выставить, если есть такая настройка, максимально возможный период хранения (ротации). Если логов нет, но нужно выполнить анализ за последние несколько дней, их можно попробовать запросить в тех поддержке хостинга. На большинстве shared-хостингов логи можно найти в директории logs, которая расположена на один или два уровня выше директории public_html (www). Итак, будем считать, что логи на хостинге есть и путь до них известен.

Подключаемся по SSH и переходим в директорию с логами веб-сервера, которые на виртуальных хостингах хранятся обычно за последние 5—7 дней. Если вывести список файлов в директории, скорее всего там будет access_log за сегодняшний день, а также access_log.1.gz, access_log.2.gz,… – это архивированные логи за предыдущие дни.

Начать анализ лога можно с запросов, которые были вы-

полнены методом POST:

grep «POST /' access_log

или

cat access_log | grep «POST /»

Результат вывода можно сохранить в новый текстовый файл для дальнейшего анализа:

grep «POST /' access_log> post_today. txt

Как сделать то же для лога, заархивированного gzip? Для этого есть команда zcat (аналогично cat, но распечатывает содержимое заархивированного файла).

zcat access_log.1.gz | grep «POST /'> post_today. txt

Для анализа подозрительной активности желательно использовать выборку по всем доступным логам. Поэтому далее в примерах будем использовать команду find, которая выполнит поиск всех файлов, а затем выполнит для каждого соответствующую команду (например, zcat).

Как определить попытки взлома или поиска уязвимых скриптов?

Например, можно найти все обращения к несуществующим скриптам. php во всех доступных логах.

grep 'php HTTP. 404» access_log*

find. -name «.gz' -exec zcat {} \; | grep 'php HTTP.* 404»*

(вместо -exec можно использовать xarg для вызова zcat.)

Еще можно поискать все неуспешные обращения к скриптам php (к которым был закрыт доступ).

find. -name «.gz' -exec zcat {} \; | grep 'php HTTP.* 403»*

Здесь мы ищем запросы, в которых встречается расширение php и статус 403.

Далее посмотрим по всем доступным логам число успешных обращений к скриптам, отсортируем их по числу обращений и выведем ТОП-50 самых популярных. Выборку сделаем в три шага: сначала выполним поиск по access_log, затем по всем access_log.*.gz, выведем результаты в файл, а затем используем его для сортировки.

find. -name «.gz' -exec zcat {} \; | grep 'php HTTP.* 200»> php. txt*

grep 'php HTTP. 200» access_log>> php. txt*

cut -d «"» -f2 php. txt | cut -d ' ' -f2 | cut -d»?» -f1 | sort | uniq -c | sort -n | tail -50

Для сайта на Wordpress результат может выглядеть так:

(примеры для Wordpress приведены исключительно для иллюстрации, в действительности описанный подход и команды не ограничиваются данной CMS. Приведенные команды можно использовать для анализа журналов веб-сервера сайтов, работающих на любых php фреймворках и системах управления (CMS), а также просто на php скриптах.)

1 /wp-admin/edit.php
1 /wp-admin/index.php
1 /wp-admin/update-core.php
1 /wp-admin/upload.php
2 /wp-admin/users.php
3 /wp-admin/plugins.php

4 /wp-includes/x3dhbbjdu.php
4 /wp-admin/profile.php
4 /wp-admin/widgets.php
38 /wp-admin/async-upload.php
58 /wp-admin/post-new.php
1635 /wp-admin/admin-ajax.php
6732 /xmlrpc.php
14652 /wp-login.php

Из результата видно, что к файлу wp-login.php было более 14000 обращений, что ненормально. Судя по всему на сайт была (или еще идет) брутфорс атака с попыткой подобрать доступ к панели администратора.

Большое число обращений к xmlrpc.php также может свидетельствовать о подозрительной активности. Например, через сайт могут атаковать (DDOS'ить) другие Wordpress сайты при наличие XML RPC Pingback Vulnerability.

Еще в списке подозрительными выглядят успешные обращения к /wp-includes/x3dhbbjdu.php, так как такого файла в стандартном Wordpress быть не может. При анализе он оказался хакерским шеллом.

Таким образом буквально за несколько секунд можно можно получить статистику по обращениям к скриптами, определить аномалии и даже найти часть хакерских скриптов без сканирования сайта.

Теперь давайте посмотрим, не было ли попыток взлома сайта. Например, поиска уязвимых скриптов или обращения

к хакерским шеллам. Найдем все запросы к файлам с расширением. php со статусом 404 Not Found:

find. -name «.gz' -exec zcat {} \; | grep 'php HTTP.* 404»> php_404.txt*

grep 'php HTTP. 404» access_log>> php_404.txt*

cut -d «"» -f2 php_404.txt | cut -d ' ' -f2 | cut -d»?» -f1 | sort | uniq -c | sort -n | tail -50

На этот раз результат может быть таким:

1 /info.php

1 /license.php

1 /media/market.php

1 /setup.php

1 /shell.php

1 /wp-admin/license.php

1 /wp-content/218.php

1 /wp-content/lib.php

1 /wp-content/plugins/dzs-videogallery/ajax.php

1 /wp-content/plugins/formcraft/file-upload/server/php/upload.php

1 /wp-content/plugins/inboundio-marketing/admin/partials/csv_uploader.php

1 /wp-content/plugins/reflex-gallery/admin/scripts/FileUploader/php.php

1 /wp-content/plugins/revslider/temp/update_extract/revslider/configs.php

1/wp-content/plugins/ultimate-product-catalogue/product-

sheets/wp-links-ompt.php
1/wp-content/plugins/wp-symposium/server/php/
fjlCFrorWUFEWB.php
1 /wp-content/plugins/wpshop/includes/ajax.php
1 /wp-content/setup.php
1 /wp-content/src.php
1 /wp-content/themes/NativeChurch/download/download.php
1 /wp-content/topnews/license.php
1 /wp-content/uploads/license.php
1 /wp-content/uploads/shwso.php
1 /wp-content/uploads/wp-admin-cache.php
1 /wp-content/uploads/wp-cache.php
1 /wp-content/uploads/wp-cmd.php
1 /wp-content/uploads/wp_config.php
1 /wp-content/wp-admin.php
1 /wp-update.php
1 /wso2.php
2 /wp-content/plugins/dzs-zoomsounds/ajax.php
2 /wp-content/plugins/hello.php
2 /wp-content/plugins/simple-ads-manager/sam-ajax-admin.php
3 /wp-content/plugins/dzs-zoomsounds/admin/upload.php
4 /2010/wp-login.php
4 /2011/wp-login.php
4 /2012/wp-login.php
4 /wp-content/plugins/wp-symposium/server/php/index.php

Как видим из результата, подобные обращения были. Кто-то «на удачу» пытался обратиться к хакерскому шеллу в каталоге предположительно уязвимого компонента Revolution Slider /wp-content/plugins/revslider/temp/update_extract/revslider/configs.php и к WSO Shell в корне сайта и к ряду других хакерских и уязвимых скриптов. К счастью, безуспешно.

С помощью тех же find / cat / zcat / grep можно получить список IP адресов, с которых эти запросы выполнялись, дату и время обращения. Но практической пользы в этом мало.

Больше пользы от выборки всех успешных POST запросов, так как это часто помогает найти хакерские скрипты.

find. -name «.gz' -exec zcat {} \; | grep «POST /.* 200»> post. txt*

grep «POST /. 200» access_log>> post. txt*

cut -d «"» -f2 post. txt | cut -d ' ' -f2 | cut -d»?» -f1 | sort | uniq -c | sort -n | tail -50

Результат может выглядеть следующим образом:

2 /contacts/
3 /wp-includes/x3dhbbjdu.php
7 /
8 /wp-admin/admin.php
38 /wp-admin/async-upload.php
394 /wp-cron.php
1626 /wp-admin/admin-ajax.php
1680 /wp-login.php/

6731 /xmlrpc.php

9042 /wp-login.php

Здесь видно много обращений к wp-login.php и xmlrpc.php, а также 3 успешных POST запроса к скрипту /wp-includes/x3dhbbjdu.php, которого не должно быть в Wordpress, то есть скорее всего это хакерский шелл.

Иногда полезно посмотреть выборку всех 403 Forbidden запросов, выполненных методом POST:

find. -name «.gz' -exec zcat {} \; | grep «POST /.* 403»> post_403.txt*

grep «POST /. 403» access_log>> post_403.txt*

cut -d «"» -f2 post_403.txt | cut -d ' ' -f2 | cut -d»?» -f1 | sort | uniq -c | sort -n | tail -50

В нашем случае это выглядело так. Не очень много, хотя это могли быть попытки эксплуатации XML RPC Pingback:

8 /xmlrpc.php

Наконец, можно выбрать TOP-50 популярных запросов к сайту за сегодня:

cut -d «"» -f2 access_log | cut -d ' ' -f2 | cut -d»?» -f1 | sort | uniq -c | sort -n | tail -50

Получаем:

6 /wp-admin/images/wordpress-logo.svg

6 /wp-admin/plugins.php

7 /wp-admin/post-new.php

8 /wp-admin/async-upload.php

9 /sitemap. xml

10 /wp-admin/users.php
13 /feed/
13 /wp-admin/
20 /wp-admin/post.php
22 /wp-admin/load-styles.php
38 /favicon. ico
52 /wp-admin/load-scripts.php
58 /wp-cron.php
71 /wp-admin/admin.php
330 /wp-admin/admin-ajax.php
1198 /
2447 /wp-login.php

Статистика обращений к /wp-login.php в access_log подтверждает, что брутфорс атака на сайт еще идет (кто-то пытается подобрать пароль), поэтому следует ограничить доступ к wp-admin по IP или с помощью серверной аутентификации, а если на сайте Wordpress нет регистрации пользователей, то можно ограничить доступ и к wp-login.php.

Таким образом без каких-либо специализированных приложений и дополнительных инструментальных средств можно быстро выполнить анализ логов веб-сервера, найти подозрительные запросы и их параметры (IP адрес, User Agent, Referer, дату/время). Все что для этого нужно – подключение по SSH и базовые навыки работы с командной строкой.

2 Лог антивирусной программы

Начинаем мы с того, что запускаем программу FTK Imager.

Журналы программы

Kaspersky Security записывает информацию о своей работе (например, сообщения об ошибках программы или предупреждения) в журнал событий Windows и в журналы событий Kaspersky Security.

О журнале событий Windows

В журнал событий Windows записывается информация о работе Kaspersky Security, на основании которой администратор Kaspersky Security или специалист по информационной безопасности могут контролировать работу программы.

События, связанные с работой Kaspersky Security, регистрируются в журнале событий Windows источником KSCM8 (службой Kaspersky Security). Базовые события, связанные с работой программы, имеют фиксированные коды событий. Вы можете использовать код события для поиска и фильтрации событий в журнале.

О журналах событий Kaspersky Security

Журналы событий Kaspersky Security представляют собой файлы формата TXT, которые хранятся локально в папке <Папка установки программы> \logs. Вы можете задать для хранения журналов другую папку.

Подробность ведения журналов событий программы зависит от установленных параметров детализации журналов.

Kaspersky Security ведет журналы событий по следующему алгоритму:

• Программа записывает информацию в конец самого нового журнала.

• Когда размер журнала достигает 100 МБ, программа архивирует его и создает новый журнал.

• По умолчанию программа хранит файлы журналов в течение 14 дней с момента внесения последнего изменения в журнал, а затем удаляет их. Вы можете установить другой срок хранения журналов.

Для каждого Сервера безопасности создаются отдельные журналы независимо от варианта развертывания программы.

В папке с журналами программы и в папке с данными программы (<Папка установки программы> \data) могут содержаться конфиденциальные данные. Программа не обеспечивает защиту от несанкционированного доступа к данным в этих папках. Вам необходимо предпринять собственные меры по защите данных в этих папках от несанкционированного доступа.

SpIDer Guard G3 файловый монитор для современных ОС Windows пришедший на замену SpIDer Guard® для старых 32 битных ОС Windows.

1. spiderg3.sys – бут-драйвер, мини-фильтр файловой си-

стемы работающий на 32- и 64 -битных ОС;

2. запускается одним из первых на начальном этапе загрузки ОС;

3. отслеживает и запоминает активность (запуск процессов, загрузку модулей, файловые манипуляции…) всех процессов с самого начала загрузки ОС;

4. блокирует доступ к файлам по запросу клиента.

Настройки SpIDer Guard G3

Все настройки SpIDer Guard G3 хранятся в реестре, в ключе

HKEY_LOCAL_MACHINE\SOFTWARE\Doctor Web \Scanning Engine\SpIDer Guard\Settings

и при изменении влияют на всю систему.

Основные настройки

Core/LicensePath= [каталог]

параметр позволяет указать каталог в котором находится лицензионный ключ. Обычно это каталог установки антивируса.

Core/LicenseFile= [путь]

параметр позоляет указать для SpIDerG3, конкретный лицензионный ключ. Параметр устарел и не используется, заменен на Core/LicensePath.

Конец ознакомительного фрагмента.