www.cyberok.ru
info@cyberok.ru
+7 (495) 137-7337

Редтимим мониторинг: рекон Grafana

Автор: Сергей Гордейчик, генеральный директор СайберОК

Введение

Совсем недавно, принимая участие в Кибериспытаниях на платформе Standoff365, команда CyberOK на этапе начального рекона без особых сложностей получила доступ к системе мониторинга Grafana заказчика.

Быстрый анализ показал, что хост с Grafana используется для сбора метрик с прода и одной ногой находится в Интернете, а второй – во внутренней сети. Такие узлы являются лакомым кусочком, поэтому, естественно, мы сделали стойку и начали рыть.

К сожалению для нас и к счастью для владельца, Grafana была достаточно свежей и не была подвержена известным уязвимостям, дающим возможность сразу «бахнуть RCE». Поэтому нам пришлось изрядно попотеть для реализации недопустимого события и разобраться в сплойтинге корпоративного мониторинга встроенными средствами. В этой статье мы расскажем об основных находках, а в дальнейших — погрузимся в детали технической реализации.

Если у вас есть свои техники постэкплуатации Grafana и корпоративного мониторинга, пишите в комментариях!

Рекон Grafana

Предположим, в ходе пентеста вы получили доступ к Grafana. Что посмотреть в первую очередь? Вот короткий чеклист.

Пользователи и их активность:

- Список пользователей.
- История входов (IP-адреса, временные метки).
- Роли пользователей (Admin, Viewer, Editor).

Датасорсы:

- Типы датасорсов (Prometheus, MySQL, PostgreSQL, Elasticsearch и т.д.).
- URL и IP-адреса датасорсов.
- Параметры аутентификации (логины, пароли, токены).

Дашборды:

- Конфигурации дашбордов.
- Запросы к датасорсам, которые могут содержать чувствительные данные, такие как API-ключи и токены доступа, ссылки на секреты или даже пароли в открытом виде.
- Данные в дашбордах. В нашем случае это были логи Prometheus, что дало очень много полезных данных о кластере Kubernetes и других элементах приложения, включая чувствительные API.

Иногда достаточно даже этого этапа. Иногда нет. Что дальше?

- Делаем grafmap

Итак, если не очень повезло, то все, что у нас есть, — это какое-то количество внутренних IP-адресов и доменных имен, имена учетных записей. А если случилось чудо — то и пароли или токены доступа. Что делать дальше?

По классике пентеста нужно максимально расширить свои знания, просканировать порты и идентифицировать доступные сервисы. Если вы уже внутри или элементы инфраструктуры доступны из внешней сети, то можно запустить nmap и радоваться, но давайте рассмотрим более сложную ситуацию, когда Grafana — единственная точка доступа во внутреннюю сеть.

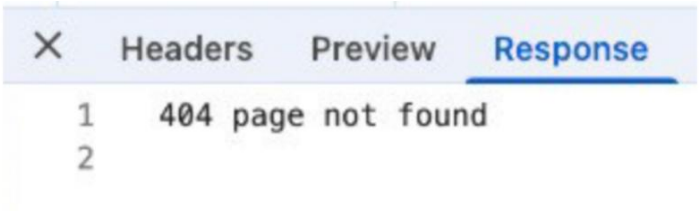
Тогда надо превратить Grafana в сканер портов! Для этого можно использовать возможность создавать и модифицировать датасорсы. В рамках нашего исследования мы использовали следующие типы расширений:

- Infinity data source plugin
- JSON plugin for Grafana

Логика достаточно простая. Мы создаем новый или модифицирум существующий источник данных, указывая в качестве цели интересующий нас IP:port или FQDN, после чего с помощью встроенной функции тестирования проверяем доступность сервиса.

Если сервис доступен, то через функцию Explore мы отправляем нужные нам запросы и анализируем результат.

Этот подход отлично работает как для HTTP(S), так и для других протоколов. Дело в том, что при тестировании плагины Grafana возвращают достаточно подробную информацию об ошибках, а в случае использования Explore можно получить полное тело ответа, включая ответы бинарных протоколов.



Пример ошибки HTTP(S)

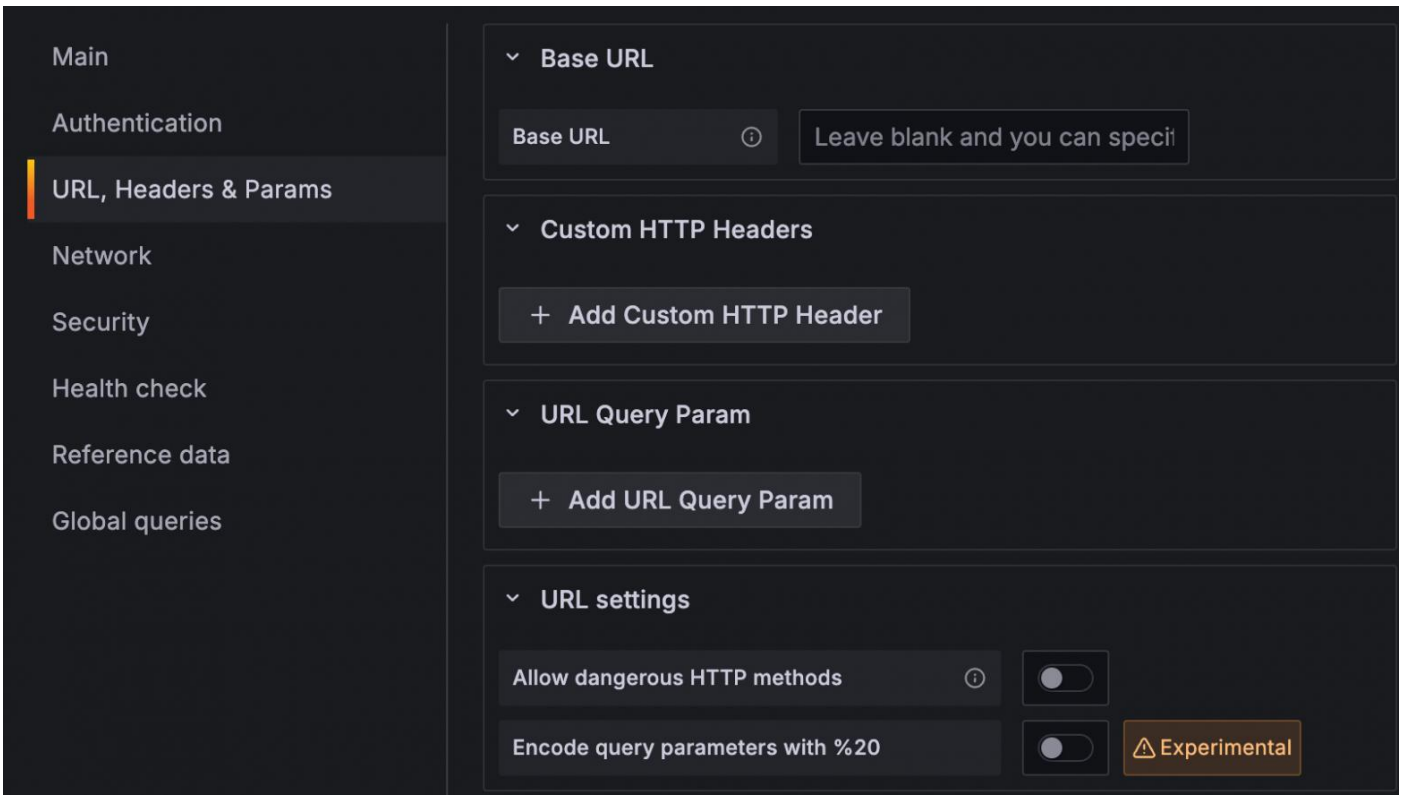
"error": "error getting response from url . no response received. Error: Get \"http://XXX.XXX.X.XX:XXX\": net/http: HTTP/1.x transport connection broken: malformed HTTP status code \"\\x00\\x00\\x00\\x04\\b\\x00\\x00\\x00\\x00\\x00\\x00\\x0f\\x00\\x01\\x00\\x00+\\a\\x00\\x00\\x00\\x00\\x00\\x7f\\xff\\xff\\xff\\x00\\x00\\x00\\x01Unexpected\""

Пример ответа бинарного протокола

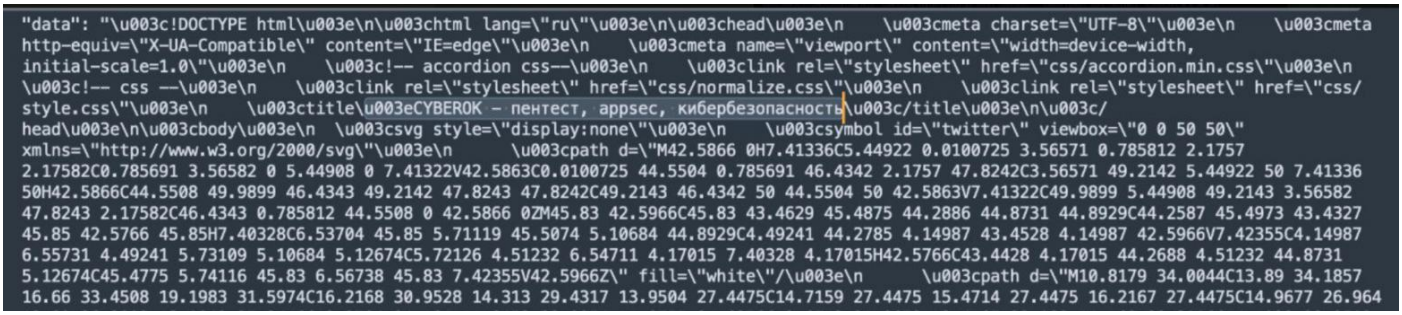
Вуаля, всё, что нам нужно далее, — это скриптофикация и база данных фингерпринтов, которую мы можем взять из Wapalyzer, Cokmap или Nmap и наш Grafmap готов. Расскажем об этом подробнее в следующей статье.

SSRF in da box

Как уже догадался проницательный читатель, Grafana — это черт в коробке готовая машинка для SSRF. Через плагин Infinity мы может отправлять произвольные HTTP(S) запросы и получать ответы. Более того, можно использовать различные методы HTTP, параметры аутентификации (Basic Auth, Bearer Token и т.д.), кастомные заголовки, параметры запроса...



И даже прокси можно указать, если вдруг у Grafana нет прямого доступа в Интернет. Чем не чудо?



Делаем SSRF

Как мы можем использовать SSRF?

1. Расширенный фингерпринт приложений. Обращаясь по разным URL и анализируя ответы, можно идентифицировать установленные приложения, их версии, параметры конфигурации.
2. Доступ к API. Например, работа с pods k8s или с интерфейсами OOB/BMC во внутренней сети.
3. Проведение произвольных веб-атак. Поскольку мы имеем возможность проводить произвольные HTTP(S) запросы, то можно искать и эксплуатировать IDOR/XXE/XPath и даже SQLi которые, к сожалению, пока еще встречаются в корпоративных веб-приложениях. В общем, вставьте ваш любимый OWASP Top X тут.

Если представить конечное развитие этой истории, то из Grafana можно сделать прокси для подключения Burp Suite или вашего любимого offensive toolkit. Конечно, это будет чуть медленней чем при нормальном шеле, но вполне рабочий вариант.

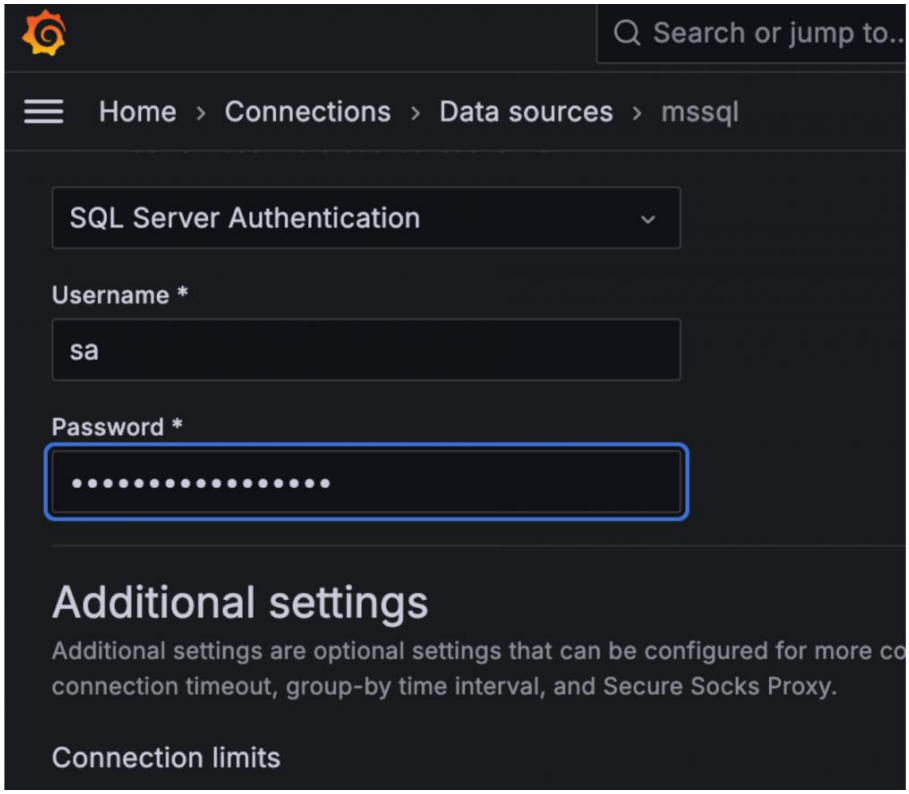
Брутфорс

Если вам удалось собрать имена пользователей, но не посчастливилось вытащить пароли или другие секреты, Grafana может помочь. Через Infinity data source plugin можно легко менять методы аутентификации и учетные записи, что позволяет спреить пароли и учетки для произвольных веб-сервисов.

Этот подход не ограничивается только паролями. Небольшая модификация дает возможность реализовать фаззинг файлов и директорий на веб серверах для поиска неочевидных объектов или ендройнтов API. Сделать этакий fuff или katana.

И это не ограничивается только веб-ресурсами. Если в ходе портскана вы нашли MS SQL, Elasticsearch или другое из десятков приложений, поддерживаемых плагинами Grafana, у вас есть прекрасная возможность для создания скриптов подбора паролей через API.

www.cyberok.ru
info@cyberok.ru
+7 (495) 137-7337



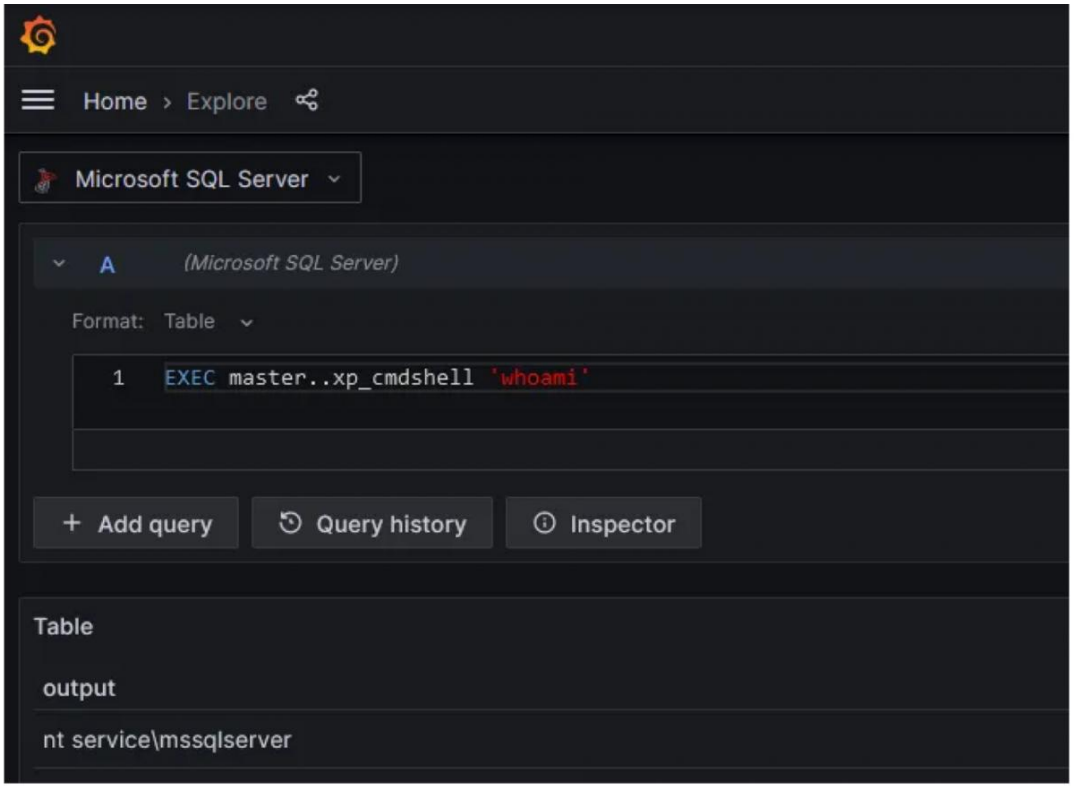
Тестируем логин mssql

SQLi from the box

Раз в год или два кто-то находит, что Grafana “уязвима для SQL Injection”. Раз, два, три.

Это достаточно спорный момент, ведь по сути многие плагины Grafana созданы для того, чтобы отправлять произвольные SQL-запросы к различным базам. С таким подходом можно говорить, что SQL Server Management Studio позволяет SQLi. Однако доля правды тут есть и неверно сконфигурированные датасорсы могут быть проблемой. Например, когда пользователь с ограниченными привилегиями имеет возможность модифицировать запросы в дашбордах. Подробнее можно узнать тут.

Однако в нашем случае мы и так имели права на модификацию датасорсов, что позволяло создавать новые соединения и выполнять произвольные запросы. Главное — знать «куда» и обладать достаточными привилегиями. Чем с большими правами Grafana обращается к серверу, тем больше возможностей у злоумышленника. Вплоть до такого.



Источник: https://medium.com/@kongi/exploiting-grafana-to-achieve-remote-command-execution-5eb0f99cb107

Это открывает много интересных возможностей, о которых мы поговорим в следующих статьях.

Резюме

Grafana, как система мониторинга и визуализации данных, является мощным инструментом для анализа состояния инфраструктуры. Однако, если злоумышленник получает доступ к Grafana, она может стать "воротами" в корпоративную сеть. В ходе нашего исследования мы выяснили, что Grafana может быть использована для:

- Сбора чувствительных данных (API-ключи, пароли, конфигурации).
- Проведения SSRF-атак.
- Сканирования портов и фингерпринта сервисов.
- Брутфорса учетных записей и подбора паролей.
- Выполнение произвольных SQL-запросов через датасорсы или SQLi через дашборды.

Эти возможности делают Grafana привлекательной целью для атакующих, особенно если она неправильно настроена или имеет избыточные права доступа. Важно понимать, как злоумышленники могут использовать Grafana для эксплуатации уязвимостей, чтобы предотвратить такие атаки.

www.cyberok.ru
info@cyberok.ru
+7 (495) 137-7337

Контрольный список для Red Team

1) Рекон и сбор информации:

- Изучите пользователей и их роли (Admin, Viewer, Editor).
- Проанализируйте датасорсы: типы, URL, параметры аутентификации.
- Просмотрите конфигурации дашбордов на наличие чувствительных данных (API-ключи, токены, ссылки на секреты).

2) Сканирование и фингерпринт:

- Используйте плагины (например, Infinity Data Source Plugin) для создания датасорсов и сканирования портов.
- Анализируйте ошибки и ответы сервисов для идентификации типов приложений.

3) SSRF-атаки:

- Отправляйте произвольные HTTP-запросы через плагины Grafana.
- Проверяйте доступность внутренних API (например, Kubernetes API, метаданные облачных провайдеров).

4) Брутфорс и подбор паролей:

- Создавайте новые датасорсы для тестирования аутентификации на различных сервисах.
- Используйте скрипты для автоматизации подбора учетных данных.

5) SQL-инъекции:

- Модифицируйте SQL-запросы в датасорсах для выполнения произвольных запросов.
- Ищите датасорсы с повышенными правами доступа.

7) Использование Grafana как прокси:

- Настройте Grafana для перенаправления запросов через Burp Suite или другие инструменты.

Контрольный список для Blue Team

1) Общие рекомендации по защите:

- **Используйте 2FA:** Внедрите двухфакторную аутентификацию для всех пользователей Grafana.
- **Ограничьте права доступа:** Разделяйте роли пользователей (Admin, Viewer, Editor) и предоставляйте минимально необходимые права.
- **Мониторинг активности:** Логируйте все действия в Grafana и анализируйте подозрительные запросы.

2) Защита датасорсов:

- **Шифруйте чувствительные данные:** Храните API-ключи, пароли и токены в защищенных хранилищах (например, HashiCorp Vault, Kubernetes Secrets).
- **Ограничьте модификацию датасорсов:** Разрешите создание и изменение датасорсов только администраторам.
- **Аудит конфигураций:** Регулярно проверяйте конфигурации датасорсов и дашбордов на наличие чувствительных данных.

3) Защита от SSRF:

- **Настройте белый список URL:** Ограничьте доступ к разрешенным доменам и IP-адресам.
- **Запретите использование частных IP-адресов:** Блокируйте запросы к внутренним ресурсам (например, 127.0.0.1, 169.254.169.254).

4) Защита от брутфорса:

- **Ограничение попыток входа:** Включите защиту от брутфорса (например, блокировка после нескольких неудачных попыток).
- **Используйте сложные пароли:** Убедитесь, что все учетные записи имеют надежные пароли.

5) Защита от SQL-инъекций:

- **Проверяйте запросы:** Используйте параметризованные запросы для предотвращения внедрения SQL-кода.
- **Ограничьте права баз данных:** Назначьте минимально необходимые права для учетных записей Grafana.

6) Обновление и патчинг:

- **Регулярно обновляйте Grafana:** Убедитесь, что используется последняя версия с исправленными уязвимостями.
- **Обновляйте плагины:** Проверяйте и обновляйте все установленные плагины.

7) Сегментация сети:

- **Изолируйте Grafana:** Убедитесь, что Grafana находится в отдельном сегменте сети с ограниченным доступом.
- **Ограничьте исходящие подключения:** Блокируйте исходящие подключения Grafana к внешним ресурсам, если это не требуется.

В настоящее время система CyberOK СКИПА отслеживает около 20к экземпляров Grafana в Рунете, 15% из них уязвимы для критических уязвимостей. Наверное, это не очень хорошо. Не дайте себя обграфанить!