Сравнение эффективности средств обнаружения уязвимостей SQLi

Петухов Андрей, Валиев Карим
Лаборатория вычислительных комплексов
Факультет ВМиК МГУ имени М.В. Ломоносова

Эффективное снижение издержек

“А можно получше, но подешевле?”

- Огромное количество сканеров: платных и бесплатных, общего назначения и специализированных
  - free: W3AF, skipfish, wapiti, arachni, Grendel-scan, secubat
    - специализированные: sqlMap, XSS Rays, SQLiX
  - $$: Acunetix, IBM AppScan, HP WebInspect, NTOSpider, Cenzic Hailstorm, и много других
- Какова область эффективного применения сканеров SQLi?
  - хочется очертить класс уязвимостей SQLi, для которых применение автоматических средств будет давать приемлемый результат
    - межмодульные уязвимости не рассматриваем
  - сэкономленное время специалисты потратят на поиск более “тонких” уязвимостей
- Действительно ли коммерческие сканеры эффективнее бесплатных?
  - даже если это так, может существует суперпозиция бесплатных сканеров, которые дают такой же результат?
- Для поиска ответов на эти вопросы мы решили создать среду для benchmark’а сканеров SQLi

Результаты тестирования

- Что: sqlMap-0.8, skipfish (Michal Zalewski & Google), wapiti, Acunetix
- На чем: LAMP
- Ответы на вопросы:
  - Acunetix не победил
    - в части обнаружения уязвимостей SQLi в веб-приложениях, работающих с MySQL, Acunetix не продемонстрировал лучший результат ни в одном классе тестов
  - Область эффективного применения: веб-приложение выводит ошибки СУБД и/или СУБД предоставляет возможность замедлить выполнение SQL-запроса (sleep)
  - Лучшая комбинация из протестированных средств skipfish + wapiti
    - даже в лучшей комбинации не удается с хорошими показателями (> 50% тестов) обнаруживать уязвимости некоторых классов
  - Почти неразрешимая проблема для протестированных сканеров - найти SQLi при условиях:
    - ошибки СУБД не выводятся в HTTP-ответ
    - в СУБД отсутствуют или запрещены функций типа sleep
    - HTTP-ответ не стабилен (см. рекламные баннеры)

Методика тестирования

Тестовое покрытие

- Задача - смоделировать как можно больше различных вариантов уязвимости SQLi
  - и не забыть про тестирование ложных срабатываний!
- Мы зафиксировали шаги, которые выполняет типичное веб-приложение, работающее с СУБД
  1. получение пользовательских данных из HTTP-запроса
  2. обработка пользовательских данных на корректность
  3. формирование и выполнение SQL-запроса
  4. получение результата SQL-запроса / обработка исключительной ситуации
  5. формирование HTTP-ответа
- Для каждого шага мы провели классификацию возможных способов его реализации
  - критерии классификации выбирались из соображений наличия специфики при обнаружении уязвимости SQLi в образующихся классах
    - например, имеет смысл классифицировать по типу запроса (INSERT/SELECT) и не имеет смысл классифицировать по типу используемых триггеров (ON INSERT/ON DELETE)
- Результирующий тестовый набор состоит из всех возможных комбинаций вариантов реализации каждого шага

Методика тестирования

Проведение испытаний

- Среда для проведения испытаний состоит из следующих элементов:
  - окружение LAMP
  - генератор тестов
    - можно легко расширить тестовое покрытие!
  - обертки для запуска сканеров и анализа их результатов
  - планировщик запуска сканеров
  - анализатор результатов
- Этапы benchmark'а:
  - генерация тестов
    - результатом является набор уязвимых и неуязвимых php-сценариев
    - сейчас генерируется 27680 тестов
  - многопоточный планировщик запускает экземпляры сканеров и контролирует их работу
  - анализатор результатов генерирует итоговый отчет, в котором проставлены метрики эффективности средств по каждому классу тестового набора
    - интересующие суперклассы, по которым анализатор осуществляет суммирование, задаются в конфигурационном файле

Дальнейшие планы

- Протестировать: W3AF, SQLiX, IBM AppScan, HP WebInspect
- Связаться с разработчиками сканеров и доложить им о результатах испытаний
- Сделать публично доступной образ виртуальной машины, на которой развернута среда для тестирования сканеров
- Получить и обработать поступивший feedback
- Расширить тестовую среду наборами для тестирования возможностей сканеров по работе с другими версиями СУБД: SQL Server, Postgres, Oracle, SQLite
- В светлом будущем: реализовать аналогичную среду для тестирования возможностей обнаружения XSS

Контактная информация

Петухов Андрей
email: petand@lvk.cs.msu.su
blog: http://andrepetukhov.wordpress.com/

Карим Валиев
email: karim@lvk.cs.msu.su

Лаборатория вычислительных комплексов
Москва, 119899, Ленинские горы вл. 1/52,
факультет ВМК МГУ имени М.В. Ломоносова,
комната 764

Related work

- Andreas Wiegenstein, Frederik Weidemann, Dr. Markus Schumacher, Sebastian Schinzel. Web Application Vulnerability Scanners – a Benchmark. Октябрь 2006.

- Larry Suto. Analyzing the Effectiveness and Coverage of Web Application Security Scanners. Октябрь 2007. И ответы на него от Ory Segal (IBM) и Jeff Forristal (HP).

- Anantasec. Web Application Scanners Comparison. Январь 2007.

- Larry Suto. Analyzing the Accuracy and Time Costs of Web Application Security Scanners. Февраль 2010. И ответы на него от Acunetix, NT Objectives, Jeremiah Grossman и HP.

- Jason Bau, Elie Bursztein, Divij Gupta, John Mitchell. State of the Art: Automated Black-Box Web Application Vulnerability Testing. Май 2010.

- Adam Doupe, Marco Cova, and Giovanni Vigna. Why Johnny Can't Pentest: An Analysis of Black-box Web Vulnerability Scanners. Июль 2010.

- Shay Chen. Web Application Scanners Accuracy Assessment. Декабрь 2010.

Спасибо за внимание!

Вопросы?