винного типу, коли сервер не може обробити величезну кількість вхідних пакетів.

При бажанні (бюджеті) можна «завалити» будь-який сервер. Обмежили кількість звернень з однієї IP-адреси? Отримайте DDOS (distributed - розподілений), коли звернення проводяться не з одного комп'ютера. DDoS використовується там, де звичайний DoS неефективний. Для цього кілька комп'ютерів об'єднуються, далі кожен створює DoS атаку на систему жертви.

Проти DOS атаки неможливо захиститися на 100%, але можна зробити обмеження на кількість спроб логіна з однієї IP-адреси в деяку кількість часу. Наприклад - не більше 5 в 10 хвилин. При вичерпанні показувати повідомлення "почекайте" або пропонувати ввести CAPTCHA. Деякі системи просять ввести CAPTCHA взагалі при кожній спробі логіна.

Список використаних джерел:

1. Лучшие PHP фреймворки в 2017 году. URL: blog.liveedu.tv/список-лучших-php-фреймворков/ (дата звернення: 21.10.2017).
2. Виды взломов сайтов и их предотвращение. URL: http://captcha.ru/articles/antihack/ (дата звернення: 10.10.2017).
3. Виды хакерских атак. URL: https://sites.google.com/site/hakerskieataki/home/vidy-hakerskih-atak (дата звернення: 10.10.2017).

Одержано 27.10.2017

УДК 343.9 : 004

Ян Андреевич ИЖБОЛДИН,
курсант факультета № 4 Харьковского национального университета внутренних дел

ОТДЕЛЬНЫЕ УЯЗВИМОСТИ В ЯДРЕ ОПЕРАЦИОННОЙ СИСТЕМЫ LINUX

Уязвимость Dirty COW (CVE-2016-5195) - серьезная программная уязвимость в ядре Linux, существующая с 2007 года и исправленная в октябре 2016 года. С её помощью локальный пользователь может повысить свои права и получить привилегии из-за ошибки состязания в реализации механизма копирования при записи для страниц памяти «dirty bit», помеченных флагом dirty.

Проблема возникает при многочисленном одновременном вызове системной функции «madvise» (MADV_DONTNEED) и

записи в страницу памяти, относительно которой пользователь не имеет права доступа и внесения изменений. Эти вызовы осуществляются из разных потоков одновременно. При попытке записи в read-only COW - страницу памяти, ядро автоматически создаёт её копию, после чего записывает данные в созданную копию. Исходная страница памяти при этом остаётся нетронутой. Код уязвимого ядра Linux не осуществляет проверку, завершен ли процесс создания копии и существует ли данная копия, прежде чем начать запись по запрашиваемому адресу памяти. Поскольку это две последующие операции, считалось, что несанкционированный доступ к ним – маловероятен.

Для реализации возможностей данной уязвимости используется эксплойт «Dirty COW», при активизации которого создаются два потока: поток A и поток B. Системный вызов madvise (MADV_DONTNEED) в потоке A сообщает ядру о том, что исполняемая программа больше не нуждается в использовании указанной страницы памяти, поэтому ядро сразу же удаляет все копии данной страницы, но оставляет доступ к ней, не внося изменений в путь к прежнему адресу.

Запись в эту же страницу из потока B приводит к необходимости создания новых копий указанной страницы памяти. При одновременном выполнении описанных выше операций, с очень малой вероятностью может произойти ситуация, когда копия страницы удаляется сразу же после её создания, но перед операцией записи. В случае возникновения неблагоприятной ситуации, ядро может сохранить данные в исходную read-only страницу памяти, а не в её копию. При многочисленном повторении запросов из разных потоков происходит перенагрузка на сервер и маловероятное событие обязательно произойдёт, в результате чего эксплойт получает право на изменение исходной read-only страницы. Обычно процесс занимает не более нескольких секунд. Необходимым условием для использования уязвимости является доступ на чтение к файлу или участку памяти.

Это означает, что локальный пользователь не может напрямую перезаписать системные файлы, которые не доступны для чтения, как например /etc/shadow, что позволило бы сменить пароль суперпользователя. Однако уязвимость позволяет записать произвольный код в любой исполняемый файл, в том числе любой suid-файл.

Таким образом, пользователь получает возможность вносить изменения в системные файлы, запускаемые им от имени суперпользователя (обладателя root прав). Например, становится возможным заменить suid-файл ping на системный терминал, который запустится от имени суперпользователя. Несмотря на то, что ошибка повышения привилегий реализуется исключительно среди локальных пользователей. Злоумышленники, не имеющие доступа к локальной сети, могут использовать уязвимость в сочетании с другими эксплойтами, которые предоставляют возможность удаленного управления и выполнение непривилегированного кода.

Использование такого сочетания инструментов не оставляет цифровых следов в системных журналах и приводит к полному взлому удаленной системы.

Список використаних джерел:

1. Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dirty COW /proc/self/mem' Race Condition Privilege Escalation (SUID Method). URL: https://www.exploit-db.com/exploits/40616/ (дата звернення: 23.10.2017).

Одержано 30.10.2017

УДК 004.492.2

Ігор Володимирович КОБЗЕВ,
кандидат технічних наук, доцент, доцент кафедри інформаційних технологій і систем управління Харківського регіонального інституту державного управління Національної академії державного управління при Президентові України

Вікторія Анатоліївна ЛУК'ЯНОВА,
кандидат педагогічних наук, завідувач кафедри природознавчих наук Харківського національного університету радіоелектроніки

МЕТОДИ ЗАХИСТУ САЙТУ НА CMS WORDPRESS

Сфера інформаційної безпеки – актуальне питання сучасності. Захист сайтів від «хакерів» стає глобальною проблемою, над вирішенням якої працюють фахівці всього світу. Бізнес, побудований в агресивному середовищі Інтернет, уразливий і схильний до нападів з боку конкурентів і недоброзичливців. Єдиного інструменту для усунення усіх загроз несанкціонованого доступу до сайтів просто не існує [1].

WordPress є популярною безкоштовною системою управління контентом для сайтів. Відкрита форма WordPress використовується вже на 28,7 % світових сайтів. При цьому на ринку систем управління контентом (CMS) WordPress є