УДК 004.49

Тасенко Д.В.
Кіровоградський національний технічний університет

Методи захисту від злому систем шифрування жорстких дисків шляхом «холодного перезапуску»

Більшість експертів вважають, що дані з оперативної пам'яті комп'ютера стираються практично миттєво після відключення живлення, або вважають, що залишкові дані вкрай складно витягти без використання спеціального обладнання. Але, ці припущення є некоректними. Звичайна DRAM пам'ять втрачає дані поступово протягом декількох секунд, навіть при звичайних температурах, а навіть якщо мікросхема пам'яті буде вилучена з материнської плати, дані збережуться в ній протягом хвилин або навіть годин, за умови зберігання цієї мікросхеми при низьких температурах. Залишкові дані можуть бути відновлені за допомогою простих методів, які вимагають короткочасного фізичного доступу до комп'ютера.

Реалізація захисту від атак на оперативну пам'ять нетривіальна, оскільки використовувані криптографічні ключі необхідно деінде зберігати. Пропонується сфокусувати зусилля на знищення або прихованні ключів до того, як зловмисник зможе отримати фізичний доступ до ПК, запобігаючи запуск ПО для дампа оперативної пам'яті, фізично захищаючи мікросхеми ОЗП і по можливості знижуючи термін зберігання даних в ОЗП.

Перезапис пам’яті**.** Насамперед, треба по-можливості уникати зберігання ключів в ОЗП. Необхідно перезаписувати ключову інформацію, якщо вона більше не використовується, і запобігати копіювання даних в файли підкачки. Пам’ять повинна очищатись завчасно засобами ОС або додаткових бібліотек. Звісно, ці міри не захистять ключі, які використовуються в даний момент, оскільки вони мають зберігатись в пам’яті, наприклад такі ключі, які використовуються для шифрування дисків чи на захищених веб-серверах. Також, ОЗП повинна очищатись в процесі завантаження. Деякі ПК можуть бути настроєні таким чином, щоб очищати ОЗП при завантаженні за допомогою очищаючого POST запита (Power-on Self-Test) до того, як завантажувати ОС. Якщо зловмисник не зможе запобігти виконання даного запиту, то на даному ПК у нього не буде можливості зробити дамп пам’яті з важливою інформацією. Але, у нього все ще лишається можливість витягнути мікросхеми ОЗП і вставити їх в інший ПК з необхідними йому настройками BIOS.

Обмеження завантаження з мережі або з переносних носіїв**.** Більшість атак даного методі реалізується з використанням завантаження по мережі або з переносних носіїв. ПК повинен бути настроєний так, щоб вимагати пароль адміністратора для завантаження з цих джерел. Але, необхідно відмітити, що навіть якщо система настроєна на завантаження тільки з основного жорсткого диску, атакуючий може змінити час жорсткий диск, або, в багатьох випадках, скинути NVRAM комп’ютера для відкату на первинні налаштування BIOS.

Безпечний сплячий режим. Результати досліджень показали, що просте блокування робочого стола ПК (тобто, ОС продовжує працювати, але, для того, щоб почати з нею взаємодію, потрібен ввід паролю) не захищає вміст ОЗП. Сплячий режим не ефективний і в тому випадку, якщо ПК блокується при повернені зі сплячого режиму, оскільки зловмисник може активувати повернення зі сплячого режиму, після чого перезавантажити ноутбук і зробити дамп пам’яті. Режим hibernate (вміст ОЗУ копіюється

на жорсткий диск) також не допоможе, крім випадків використання ключової інформації на відчужуваних для відновлення нормального функціонування. В більшості систем шифрування жорстких дисків, користувачі можуть захиститись виключенням ПК. (Система Bitlocker в базовому режимі роботи TPM модуля лишається вразливою, оскільки диск буде підключений автоматично, коли ПК буде ввімкнений). Вміст пам'яті може зберігатися протягом короткого періоду після вимкнення, тому рекомендується поспостерігати за своєю робочою станцією ще протягом пари хвилин. Не дивлячись на свою ефективність, дана міра вкрай незручна в зв'язку з довгим навантаженням робочих станцій. Перехід в сплячий режим можна убезпечити наступними способами: вимагати пароль чи інакший інший секрет, аби «розбудити» робочу станцію і шифрувати вміст пам'яті ключом, похідним від цього пароля. Пароль має бути стійким, так як зловмисник може зробити дамп пам'яті і після чого спробувати підібрати пароль перебором. Якщо ж шифрування всієї пам'яті неможливо, необхідно шифрувати тільки ті області, які містять ключову інформацію. Деякі системи можуть бути налаштовані таким чином, щоб переходити в такий тип захищеного сплячого режиму, хоча це зазвичай і не є налаштуванням за замовчуванням.

Відмова від попередніх обчислень**.** Наші дослідження показали, що використання попередніх обчислень для того, щоб прискорити криптографічні операції робить ключову інформацію більш вразливою. Попередні обчислення призводять до того, що в пам'яті з'являється надлишкова інформації про ключових даних, що дозволяє зловмисникові відновити ключі навіть в разі наявності помилок. Відмова від попередніх обчислень знизить продуктивність, оскільки потенційно складні обчислення доведеться повторювати. Але, наприклад, можна кешувати попередньо вирахувані значення на певний проміжок часу і стирати отримані дані, якщо вони не використовуються протягом цього інтервалу. Такий підхід є компромісом між безпекою та продуктивністю системи.

Розширення ключів. Інший спосіб запобігти відновлення ключів - це зміна ключової інформації, що зберігається в пам'яті, таким чином, щоб ускладнити відновлення ключа через різні помилки. Цей метод був розглянутий в теорії, де була показана функція, стійка до розкриття, чиї вхідні дані залишаються прихованими, навіть якщо практично всі вихідні дані були виявлені, що дуже схоже на роботу односпрямованих функцій. На практиці, уявіть, що у нас є 256-бітний AES ключ K, який в даний момент не використовується, але знадобиться пізніше. Ми не можемо перезаписати його, але ми хочемо зробити його стійким до спроб відновлення. Один із способів досягнення цього - це виділити велику B-бітну область даних, заповнити її випадковими даними R, після чого зберігати в пам'яті результат наступного перетворення K + H (R), де H - це хеш функція, наприклад SHA-256. Тепер уявіть, що електрика була відключена, це призведе до того, що d біт в даній області будуть змінені. Якщо хеш функція стійка, при спробі відновлення ключа K, зловмисник може розраховувати тільки на те, що він зможе вгадати які біти області B були змінені з приблизно половини, які могли змінитися. Якщо d біт були змінені, зловмисникові доведеться провести пошук області розміром (B / 2 + d) / d щоб знайти коректні значення R і вже після цього відновити ключ K. Якщо область B велика, такий пошук може бути дуже довгий, навіть якщо d відносно мала. Теоретично, таким способом можна зберігати всі ключі, розраховуючи кожен ключ, тільки коли це нам необхідно, і видаляючи його, коли він нам не потрібен. Таким чином, застосовуючи вищеописаний метод, ми може зберігати ключі в пам'яті.

Фізичний захист. Деякі подібних атак основувалися на наявності фізичного доступу до мікросхем пам'яті. Такі атаки можуть бути попереджені фізичним захистом

пам'яті. Наприклад, модулі пам'яті знаходяться в закритому корпусі ПК, або залиті епоксидним клеєм, щоб запобігти спробам їх вилучення або доступу до них. Так само, можна реалізувати затирання пам'яті як відповідну реакцію на низькі температури або спроби відкрити корпус. Такий спосіб потребує установки датчиків з незалежною системою живлення. Багато з таких способів пов'язані з апаратурою, захищеної від несанкціонованого втручання і можуть сильно підвищити вартість робочої станції. З іншого боку, використання пам'яті, припаяної до материнської плати, обійдеться набагато дешевше.

Зміна архітектури. Можна змінити архітектуру ПК. Що неможливо для ПК які використовуються, зате дозволить убезпечити нові. Перший підхід полягає в тому, щоб спроектувати DRAM модулі таким чином, щоб вони швидше стирали всі дані. Це може бути непросто, оскільки мета якомога швидшого стирання даних, суперечить іншій меті, аби дані не пропадали між періодами поновлення пам'яті. Інший підхід полягає в додаванні апаратури зберігання ключової інформації, яка б гарантовано стирала всю інформацію зі своїх сховищ при запуску, перезапуску і виключенні. Таким чином, ми отримаємо надійне місце для зберігання декількох ключів, хоча вразливість, пов'язана з їх попередніми обчисленнями залишиться. Інші експерти запропонували архітектуру, в рамках якої вміст пам'яті буде постійно шифруватися. Якщо, плюс до цього, реалізувати стирання ключів при перезавантаженні і відключенні електрики, то даний спосіб забезпечить достатню захищеність від подібних атак.

Довірені обчислення**.** Апаратура, відповідна концепції «довірених обчислень», наприклад, у вигляді TPM модулів вже використовується в деяких ПК. Незважаючи на свою корисність в захисті від деяких атак, в своїй нинішній формі таке обладнання не допомагає запобігти подібні атаки. Використовувані TPM модулі не реалізують повне шифрування. Замість цього, вони спостерігають за процесом завантаження для прийняття рішення про те, чи безпечно завантажувати ключ в ОЗУ чи ні. Якщо ПЗ необхідно використовувати ключ, то можна реалізувати наступну технологію: ключ, в придатній для використання формі не буде зберігатися в ОЗУ, до тих пір поки процес завантаження не пройде по очікуваному сценарієм. Але, як тільки ключ виявляється в оперативній пам'яті - він відразу становиться мішенню для атак. TPM модулі можуть запобігти завантаження ключа в пам'ять, але вони не запобігають його зчитування з пам'яті.

Схоже, що немає простого способу усунути знайдені вразливості. Зміна ПО швидше за все не буде ефективним; апаратні зміни допоможуть, але тимчасові і ресурсні витрати будуть великі; технологія «довірених обчислень» в її сьогоднішньої формі так само мало ефективна, оскільки вона не може захистити ключі знаходяться в пам'яті.

Найбільше даному ризику схильні ноутбуки, які часто знаходяться в громадських місцях і функціонують в режимах вразливих для даних атак. Наявність таких ризиків, показує, що шифрування дисків здійснює захист важливих даних в меншому ступені, ніж прийнято вважати.

У підсумку, можливо, доведеться розглядати DRAM пам'ять що не довірену компоненту сучасного ПК, і уникати обробки важливої конфіденційної інформації в ній. Але на даний момент це недоцільно, до тих пір, поки архітектура сучасних ПК не зміниться, щоб дозволити ПО зберігати ключі в безпечному місці.

Список використаних джерел

1. Proc. 17th USENIX Security Symposium (Sec ’08), San Jose, CA, July 2008.