23. *Диференціальні* перетворення для комп'ютерного моделювання керуючих систем: [навч. посібн. для студ. вищ. навч. закл.] / О.І. Стасюк, В.Л. Баранов, Г.Л. Баранов, О.Г. Фролова. – К.: КУЕТТ, 2005. – 135 с.

УДК 681.3

Дудикевич В.Б., Піскозуб А.З., Тимошик Н.П., Тимошик Р.П., Дуткевич Т.В.
Национальный университет "Львівська політехніка"

МЕТОДИ ТА ЗАСОБИ АНАЛІЗУ СИСТЕМИ-ПРИМАНКИ В ПРОЦЕСІ ЗЛАМУ

Підхід до отримання сигнатур від систем приманок залежить від знань про методи компрометації робочих систем та процедур аналізу інцидентів, які включають в себе порівняння еталонної операційної системи (ОС) з її скомпрометованою версією. Для цього нам необхідно розглянути можливі методики зламу чи компрометації систем та у відповідності до них сформувати власну процедуру для розслідування інцидентів в системах-приманках, процесів та причин успішного проникнення, а також можливостей моніторингу критичних областей, завдяки яким проникнення може мати успіх.

Ідея використання системи приманок дуже зручна для аналітиків безпеки завдяки наступній особливості: в спеціально підготовлену, привабливу для зловмисника систему (наприклад, система може містити якусь відому або невідому досі вразливість) потрапляє кваліфікований зловмисник. Завдяки спеціалізованому програмному забезпеченню існує можливість стеження за діями зловмисника на приманці та отримання інформації про найновіші методи, засоби та підходи в реалізації атак та шкідливого впливу (malware) на комп'ютерні системи та мережі [1].

Метою даної статті є систематизація методів та засобів аналізу систем-приманок в процесі зламу комп'ютерних систем чи мереж та рекомендації щодо організації розслідування зламу, вибору засобів збору та аналізу подій.

Таким чином, завдяки правильному розгортанню та конфігурації системи-приманки таке проникнення може бути проаналізоване і відповідно може бути створене відповідне правило, за допомогою якого система виявлення атак зможе зупинити подібну атаку на важливу робочу систему чи мережу [2]. Для цього рекомендується використовувати спеціалізовані приманки, тобто розділяти системи-приманки на складові компоненти, наприклад: приманка з надійно захищеною ОС, але вразливим сервісом (для прикладу приманка лише з незахищеним веб-сервером чи приманка з вразливою СУБД); приманка з оновленою та захищеною ОС [3]. Для зручності аналізу атак та даних, які необхідно аналізувати, атаки на прикладні сервіси можна розбити на такі класи як атаки на сервіси та атаки на операційні системи.

Атаками на сервіси можна вважати:

- атаки відмови в обслуговуванні сервісу (DoS, DDoS);
- атаки з використанням вразливості сервісу (веб, dns, поштовий сервіс) з метою отримання принаймні мінімального локального доступу до системи або читання конфігурації системи (файлів паролів, конфігурації системи, параметрів фаєрволів);
- атаки з використанням неправильного конфігурування сервісу (firewall, privilegies overflow, php-injection, не встановлені chroot обмеження, списки контролю доступу (ACL), SELinux);
- атаки на бази даних з метою викрадення або зміни інформації (зміна вмісту веб-сторінки, перенаправлення на підставні ресурси та викрадення конфіденційної інформації, SQL Injection).

Друга категорія атак – атаки на операційні системи, є доволі поширена і причинами її успішності є:

- неправильна реалізація мережевої взаємодії, контролю трафіку та мережного стеку;

- помилки в ядрі операційної системи;
- некоректна організація доступу до файлових систем та системних ресурсів;
- розгортання непотрібних мережевих сервісів;
- некоректна реалізація прикладного програмного забезпечення (ПЗ) користувачів.

Загалом, атака має на меті або отримання безпосереднього доступу в систему, або її експлуатацію в інших шкідливих цілях. Також не можна забувати про автентичність системи-приманки та її "привабливість".

Підходи для організації систем-приманок

Аналіз інциденту розпочинається від першої сигналізації про зміну стану чи параметрів системи, оскільки система-приманка – це спеціалізована ОС та сервіс, на якій жоден легітимний користувач не проводить ніяких дій. Таким чином, поява будь-якої взаємодії з такою системою може розцінюватись як початок атаки [2].

У випадку, якщо зловмисника цікавить використання ресурсів вашого сервера для організації розповсюдження malware, можна використати цю інформацію для організації ефективної системи блокування спаму, вірусів, підставних сторінок та іншого шкідливого мережевого коду, оскільки буде доступна детальна інформація про походження та зміст malware.

Від початку атаки до її завершення (завершенням атаки будемо вважати тривале припинення активності зловмисника після успішного проникнення) необхідно зібрати дані з приманки та мережі. Зазвичай процедура аналізу проводиться в автономному режимі, коли систему ізольовано, та жодних зовнішніх змін в системі неможливо завдати. Інший підхід до аналізу послідовності атаки полягає в безпосередньому стеженні за подіями на приманці в режимі реального часу – командами та засобами, якими зловмисник користується для реалізації зламу, та їх впливом на робоче середовище приманки.

Важливим залишається питання про те, що робити з приманкою, коли злам проведено успішно і всі необхідні дані збережені. У випадку, якщо відразу відновити систему до попереднього стану, зловмисник може запідозрити небезпеку та покинути приманку. Тому рекомендується організувати систему ротації приманок та їхніх адрес з метою збереження автентичності мережі приманок та організувати кілька варіантів внутрішньої будови мереж приманок для забезпечення неповторності та автентичності останніх [3]. Досвідчений зловмисник може настільки заплутати сліди, що здійснення аналізу та відтворення подій стане неможливим.

Ідеологічно системи-приманки повинні містити якусь вразливість, щоб приваблювати зловмисника, але для отримання 0-day сигнатур атак, ми повинні слідкувати за станом оновлень такої системи [4]. Пропонована нами методика полягає в тому, що для зловмисника можна організувати структуру гетерогенних мереж систем-приманок, схожу на відому гру "Хакер", де йому доведеться пройти весь шлях зламу від найпростіших і потенційно вразливих систем, до дійсно захищених. Можливо, для зламу саме таких, найбільш захищених систем він запросить своїх більш компетентних колег. Це дозволить на такій системі мереж приманок вивчати не лише найновіший інструментарій та методологію зловмисників, але й тактико-психологічну послідовність дій на атакованій системі.

Ще однією пропонованою нами методикою в даній статті, завдяки якій покращується можливість відслідковування зловмисника є "ефект голуба", суть якого полягає в тому, що на системі-приманці спеціально залишається цікавий для зловмисника "подарунок" (наприклад "Фінансова звітність за 2008 рік" в веб-форматі), при відкритті якого відбувається звернення (наприклад методом Cross Site Scripting, або Code Injection) до спеціалізованого сервера, який і реєструє автентичну IP-адресу та місцезнаходження зловмисника. А оскільки таке звернення можна сформувати унікальним для кожної версії приманки, то і локалізація зловмисника стає простішою.

Також заслуговує уваги варіант примусового зниження швидкості каналів між зловмисником і приманкою та приманкою – приманкою. З іншого боку, це може викликати додатковий об'єм даних, що може ускладнити аналіз скомпрометованої системи в цілому [4].

Ще однією з важливих рекомендацій для використання високо-інтерактивних систем-приманок є використання мінімізованих за розмірами та наповненням ОС, оскільки менша кількість файлів зменшує час аналізу та підвищує продуктивність.

Рекомендації щодо організації розслідування зламу, вибору засобів збору та аналізу подій

Наведемо декілька правил, які слід виконувати при аналізі системи:

- Ніколи не проводити аналіз систем на її оригіналі, оскільки це судовий доказ, який за жодних умов не повинен зазнавати змін. Аналіз слід проводити лише на ідентичній копії;
- Аналіз повинен проводитися в такому режимі, щоб не допустити жодних змін в системі (не пошкодити або модифікувати сліди злочину);
- Аналіз слід проводити лише перевіреними засобами, а не засобами скомпрометованої системи;
- Слід документувати всі дії та висновки при аналізі;
- Не допускати втрати даних після перезавантаження.

Комплексний розгляд та дослідження комп'ютерної системи до та після атаки повинен передбачати аналіз [5]:

- файлової системи:
  - аналіз модифікованих конфігураційних файлів;
  - аналіз модифікованих виконуваних та системних файлів, бібліотек, вихідних кодів та скриптів запуску сервісів;
- баз даних (аналіз введених команд, порівняння таблиць);
- журналів реєстрації подій (дублювання записів в журналі через захищений канал на сервер централізованого збору журналів подій, оскільки зазвичай зловмисник чистить за собою такі записи);
- процесів в пам'яті системи (систем прихованого лазу (backdoors), вірусів, скриптів);
- змін в профілях системних користувачів та їх привілеїв;
- змін процесів та методів віддаленого доступу (контроль ssh ключів доступу, авторизаційних ключів та цифрових відбитків знайомих хостів);
- стану фаєрволу;
- мережевого трафіку (при наявності можливості, наприклад сніффінг нешифрованого трафіку);
- модулів та стану ядра ОС, завантажувальних областей;
- результатів моніторингу системою виявлення атак;
- ключів реєстру, ключів запуску та параметрів робочого середовища користувачів;
- введених команд;
- відновлених файлів з файлових систем;
- виявлених каналів віддаленого керування системою з боку зловмисника (IRC, UDP sniffer).

Доцільно проводити періодичні "миттєві відбитки" системи (snapshots) разом з повною копією хостової системи. Це дасть можливість розділити атаку на логічні етапи.

Резервне копіювання системи можна реалізувати на базі drbd [6], або інших засобів мережевого дублювання інформації в режимі реального часу, з можливостями резервного копіювання та відновлення.

Засобами, якими, на нашу думку, доцільно виконувати аналіз інциденту та прослідковування процесу зламу є :

- Sebek [7]– для перехоплення всіх подій на системі приманці ще до за шифрування;
- Аналіз змін файлової системи: Aide [8], tripware [9], diff ;
- Аналіз стану памяті та ядра ОС: module-hunter [10], chkrootkits[11], fuser, lsof;
- Аналіз трафіку: tcpdump [12], Wireshark [13], netstat;
- Системи виявлення атак: snort [14], prelude [15], Cisco IDS [16], Cisco ASA [17];
- Тестування на проникність: nmap [18], Nessus [19], Core Impact [20];
- Аналіз вірусної активності: clamav [21], Kaspersky Internet Security [22] тощо;
- Готова система-приманка на базі Fedora Core - Honeywall CDROM [23].

Візуальний аналіз даних, на нашу думку, доцільно здійснювати за допомогою Honeysnap (основний засіб аналізу інциденту з записаних фрагментів мережевого трафіку в форматі pcap, включаючи IRC комунікації) [24] та Capture BAT (засіб поведінкового аналізу ПЗ для Windows ОС – контролює стан системи і процесів під час виконання аплікацій, низькорівневу активність ядра) [25].

Найпопулярнішим та використовуваним засобом аналізу систем приманок залишається backdoor Sebek [4], який дозволяє збирати всю інтерактивну інформацію між зловмисником та приманкою ще до здійснення процесу зашифровування. Sebek тісно інтегрується з ядром, що дозволяє перехоплювати більшість системних викликів, включаючи натискання клавіш, та передавати їх прихованим мережевим каналом зв'язку на сервер збору інформації. Попри всі свої досягнення, Sebek має ряд технологічних недоліків, які явно обмежують його використання для систем приманок [26].

Висновки

Резюмуючи вищесказане, хочемо зауважити, що в даний момент не існує уніфікованого та спеціалізованого продукту чи комплексу для проведення комплексного аналізу. Більшість комерційних організацій, які спеціалізуються на подібних рішеннях, займаються збором відомостей з різних джерел про результати атак, поширення спаму, malware тощо, та на комерційній основі розповсюджують правила, характеристики та ознаки атак та malware.

В статті пропонуються рекомендації щодо організації розслідування зламу, вибору засобів збору та аналізу подій. Пропонуються також нові підходи для організації систем-приманок, які дають можливість підвищити їх ефективність як засобів захисту в комп'ютерних системах та мережах.

Література

1. The Honeynet Project. http://www.honeynet.org
2. Дудикевич В.Б., Піскозуб А.З.,Тимошик Н.П., Дуткевич Т.В. "Використання віртуалізації для виявлення 0-day атак та розгортання систем віртуальних приманок". III Всеукраїнська науково-практична конференція "Інформаційні технології і безпека в управлінні", Крим, м.Севастополь, Вісник Східноукраїнського національного університету ім.. В.Даля №5 (111),2007, Ч.1. с.53-58.
3. Тимошик Н.П., Захист комп'ютерних мереж на основі техтологій Intrusion Prevention Systems + Honeynets. Збірник праць "Комп'ютерні науки та інженерія - 2006" (CSE-2006), НУЛП, 2006, с. 76-80.
4. Матвеев Д. План действий после атаки на ваш хост – расследование, восстановление, защита // Сетевые решения, 2002. – № 11 – С. 25 – 35.
5. Taras Dutkevych, Andrian Piskozub , Nazar Tymoshyk "Real-Time Intrusion Prevention and Anomaly Analyze System for Corporate Networks" IEEE International Workshop on

Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications 6-8 September 2007, Dortmund, Germany p.599-602
6. "What is DRBD" www.drbd.org
7. Sebek. http://www.honeynet.org/tools/sebek
8. "AIDE - Advanced Intrusion Detection Environment". http://www.cs.tut.fi/~rammer/aide.html
9. "Tripwire® software is a security and data integrity tool". http://sourceforge.net/projects/tripwire/
10. Clam AntiVirus. www.clamav.net/
11. Linenoise. http://www.phrack.com/issues.html?issue=61&id=3
12. Tcpdump www.tcpdump.org/
13. Wireshark. www.wireshark.org
14. Snort - the de facto standard for intrusion detection/prevention. www.snort.org/
15. Prelude IDS. www.prelude-ids.com/
16. Cisco IDS. http://www.informit.com/articles/article.aspx?p=24696
17. Cisco ASA. http://www.cisco-systems.ru/katalog/cisco-asa_51/
18. Nmap - Free Security Scanner For Network Exploration & Security ... nmap.org
19. Nessus Security scanner for Oracle and various flavors of Unix. www.nessus.org
20. Core Impact. www.coresecurity.com/
21. Clam AntiVirus. www.clamav.net
22. Kaspersky Internet Security. http://kaspersky-antivirus.kiev.ua/products/inetsecurity.htm
23. Honeywall CDROM. https://projects.honeynet.org/honeywall/
24. Honeysnap. https://projects.honeynet.org/honeysnap
25. Capture BAT. https://public.honeynet.org/mailman/listinfo/capture-bat
26. M. Dornseif, T. Holz, and C. Klein. NoSEBrEaK - Attacking Honeynets. Proc. of the 5th Annual IEEE Information Assurance Workshop, Westpoint, June 2004.

УДК 336.717:004.78

Кузнецов А.А, Евсеев С.П., Король О. Г.
Харьковский национальный экономический университет

ИССЛЕДОВАНИЕ КРИПТОГРАФИЧЕСКИХ СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ В ПЛАТЕЖНЫХ СИСТЕМАХ БАНКОВ УКРАИНЫ

Постановка проблемы в общем виде и анализ литературы. Бурное развитие компьютерных систем и сетей привело к возникновению принципиально новых, так называемых, информационно - телекоммуникационных технологий, что позволило расширить услуги платежных систем коммерческих банков (ВПС КБ), внедрить новые сервисы и технологии. Вместе с тем, интенсивное использование компьютерных систем и сетей в ВПС КБ привело к появлению новых видов компьютерного терроризма: неправомерного искажения или фальсификации банковской информации, уничтожению или разглашению определенной ее части, дезорганизации процессов обработки и передачи данных. Жизненно важные интересы субъектов ВПС КБ заключаются в том, чтобы определенная часть информации, касающаяся их безопасности, экономических и других сторон деятельности, конфиденциальная коммерческая и персональная информация, была бы постоянно доступна и в тоже время надежно защищена от неправомерного ее использования [3, 8, 9]. В работах [3, 10, 14] рассмотрены основные виды угроз в информационно-коммуникационных системах, а также механизмы обеспечения аутентичности и целостности передаваемых данных.