ISSN 2306-1561
Automation and Control in Technical Systems (ACTS)
2014, No 1.1(8), pp. 35-39.
DOI: 10.12731/2306-1561-2014-1-4

Exploits as alternative of safety of programs

Kuftinova Natal'ya Grigor'evna

Russian Federation, Ph. D., Associate Professor, Department of «Automated Control Systems».

Moscow Automobile & Road construction State Technical University, 125319, Russian Federation, Moscow, Leningradsky prospekt, 64. Tel.: +7 (499) 151-64-12. http://www.madi.ru

nat.gk@mail.ru

Abstract. In this article the question of introduction of the "harmful" program in local and remote appendices of the program is considered. The example of a shell-code of an exploit in a combination of attacks to system files is given.

Keywords: exploit, program, shell-code, protocols, web-server.

ISSN 2306-1561
Автоматизация и управление в технических системах (АУТС)
2014. – №1.1(8). – С. 35-39.
DOI: 10.12731/2306-1561-2014-1-4

УДК 004.9

Эксплойты как альтернатива безопасности программ

Куфтинова Наталья Григорьевна

Российская Федерация, кандидат технических наук, доцент кафедры «Автоматизированные системы управления».

ФГБОУ ВПО «Московский автомобильно-дорожный государственный технический университет (МАДИ)», 125319, Российская Федерация, г. Москва, Ленинградский проспект, д.64, Тел.: +7 (499) 151-64-12, http://www.madi.ru

nat.gk@mail.ru

Аннотация. В статье рассматривается вопрос внедрения «вредоносной» программы в локальные и удаленные приложения. Приводится пример shell-кода эксплойта в сочетании атак на системные файлы.

Ключевые слова: эксплойт, программа, shell-код, протоколы, web-сервер.

1. Введение

Эксплойт, эксплоит, сплоит (англ. exploit, эксплуатировать) - компьютерная программа, фрагмент программного кода или последовательность команд, использующие уязвимости в программном обеспечении и применяемые для проведения атаки на вычислительную систему [1]. Целью атаки может быть как захват контроля над системой (повышение привилегий), так и нарушение её функционирования (DoS-атака).

Написание полнофункционального эксплойта - это целостная задача не только для программирования локальных и удаленных приложений, но и для атаки на уязвимость программы. Как правило, любой эксплойт содержит shell-код, который запускает интерактивную оболочку, модифицирует системные файлы или открывает в режиме просмотра порт, открывая при этом возможности внедрения «вредоносной» программы. Уязвимости, связанные с дефектами протоколов могут привести к неисправности приложений программы на прикладном уровне [4].

2. Поиск уязвимостей в программе

Чтобы отыскивать слабые с точки зрения безопасности места в программах и писать для них эксплойты, нужно понимать, какие бывают уязвимости. Их можно отнести к нескольким категориям: ошибки связанные с форматными строками и переполнение буфера. Методы поиска новых уязвимостей включают отыскание некорректного кода в исходном тексте, отправку неожиданных данных приложению и изучение программы на предмет наличия логических ошибок. В процессе поиска уязвимости нужно обращать внимание на различные аспекты [2, 3]:

Доступен ли исходный текст?

Сколько людей уже знакомились с исходным текстом и кто эти люди?

Имеет ли смысл тратить силы на автоматизированное генерирование случайных исходных данных для программы?

Сколько времени потребуется для организации тестовой среды?

Умение писать эксплойты ценно как для исследователей, так и для конечных пользователей. Узнать о состоянии дел с известными уязвимостями можно, отслеживая протоколы изменений в онлайновой системе управления версиями (CVS), если речь идет о программах с открытыми исходными текстами [4 – 8]. Авторы пакетов OpenSSL, OpenSSH, FreeBSD и OpenBSD исправляли обнаруженные ошибки, фиксируя их в CVS, еще до того как информация об уязвимостях была опубликована. Остановив свой выбор на каком-то приложении проверьте все или, по крайней мере, наиболее распространенные классы ошибок, например, переполнение стека, затирание кучи, атаки на форматную строку, переполнение целых чисел. Примите во внимание то, сколько времени приложение уже известно и сколько в нем было обнаружено ошибок ранее. Если число их невелико, то к каким классам они принадлежат? Например, если известны лишь ошибки из-за переполнения стека, поищите что-нибудь, связанное с целыми числами, поскольку те, кто исследовал программу раньше, скорее всего, обращали внимание только на самые легкие для обнаружения проблемы. Проглядите также отчеты об ошибках, найденных в конкурирующих приложениях; возможно, что они имеют схожие уязвимости.

Дав представление о методике обнаружения уязвимостей, перейдем к самим эксплойтам. Начнем с вопроса о том, как применяются эксплойты для атаки на локальные и удаленные программы [4].

Уязвимости могут существовать как в сетевом приложении, например, в Web-сервере, так и в локальном, скажем, в административной утилите. Хотя обычно уязвимости в локальных и удаленных программах не пересекаются, но иногда приходится взламывать их последовательно, чтобы повысить свои привилегии, так как чаще всего сетевые службы не запускаются от имени привилегированного пользователя, каковым является, скажем, root или SYSTEM. Например, такие службы, как Apache, IIS и OpenSSH работают от имени непривилегированных пользователей с очень ограниченными правами, чтобы снизить ущерб от возможного взлома. Чтобы повысить свои привилегии вслед за успешной атакой, необходимо выполнить еще и локальный эксплойт.

3. Внедрение «вредоносной» программы в удаленные приложения

Предположим, к примеру, что удалось взломать Web-сервер Apache. Тогда программист атакующий получит права пользователя apache, www или подобное (в зависимости от системы), но не root. Эксплуатация ошибок в локально установленных программах, в ядре и прочем позволит повысить привилегии до уровня root. А уж тогда атакующий получает полную свободу действий во взломанной системе.

Дистанционная атака на недавно обнаруженную уязвимость в Apache на платформе OpenBSD дает непривилегированный доступ, но в сочетании с атакой на ошибку в ядре (переполнение в системном вызове select) можно получить права root. Ниже приведен листинг команд, демонстрирующий сочетание атак на локальные и удаленные программы. На первом шаге используется переполнение кучи в Sun Solaris. Большинство удаленных уязвимостей эксплуатировать не так просто, но эта пролагает пути к аномально простому повышению привилегий.

Дистанционная атака на переполнение кучи в сервере telnefd на платформе Sun Solaris:

```
% telnet
telnet> environ define TTYPROMPT abcdef
telnet> open localhost
bin ccccccccccccccccccccccccccccccc ccccccccccccccccccccc
$ whoami
bin
```

Локальное повышение привилегий до уровня root на платформе Sun Solaris:

```
% grep dtspcd /etc/inetd.conf
dtspcd stream tcp wait root /usr/dt/dtspcd dtspcd
% Is -1 /usr/dt/dtspcd
-rwxrwxr-x   root   bin   20082 Jun 26 1999 /usr/dt/dtspcd
% cp /usr/dt/dtspcd /usr/dt/dtspcd2
% rm /usr/dt/dtspcd
% cp /bin/sh /usr/dt/dtspcd
% telnet localhost 6112
Trying 127.0.0.1...
Connected to localhost
Escape character is ' ^] ' .
id;
uid=0 (root) gid=0 (root)
```

После переполнения буфера, произошедшего в результате выполнения команд в строках 1-6, атакующий получает права пользователя user и группы bin. Так как файл /usr/dt/dtspcd доступен для записи всем членам группы bin, то атакующий может его

модифицировать. Интересно, однако, что эта программа вызывается inetd и, следовательно, работает от имени root. Тога атакующий делает копию исходного файла dtspcd, а затем подменяет его файлом /bin/sh. Теперь с помощью telnet программист соединяется с портом 6112, на котором работает dtspcd, и получает привилегии root. Выполнив команду id (завершаемую символом ';').

4. Заключение

Таким образом, информация, полученная в результате обнаружения уязвимости, может быть использована как для написания эксплойта, так и для устранения уязвимости [1]. Поэтому в ней одинаково заинтересованы обе стороны - и взломщик, и производитель взламываемого программного обеспечения. Характер распространения этой информации определяет время, которое требуется разработчику до выпуска заплатки.

После закрытия уязвимости производителем шанс успешного применения эксплойта начинает стремительно уменьшаться. Поэтому особой популярностью среди хакеров пользуются так называемые 0day эксплойты, использующие недавно появившиеся уязвимости, которые еще не стали известны общественности.

Список информационных источников

[1] Эксплойт. [Электронный ресурс]: http://ru.wikipedia.org.

[2] Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: Учеб. пособие для вузов по спец."Вычисл. машины, комплексы, системы и сети", "Автоматизир. машины, комплексы, системы и сети", и др. / В.Г. Олифер, Н.А. Олифер. – 2-е изд. – СПб.: Питер, 2005 . – 863 с.

[3] Борисов Н.А., Лукин А.А. Информационные компьютерные сети: учеб.-метод. пособие для практ. занятий. – М.: ИМПЭ им. А.С. Грибоедова, 2002. – 63 с.

[4] Антонов П.Д. User Is A Great Obstacle For Security Systems / П.Д. Антонов, А.В. Остроух // Молодой ученый. - 2011. - №4. Т.3. - С. 62-63.

[5] Остроух А.В. Информационные технологии в научной и производственной деятельности / [ред. А.В. Остроух] - М: ООО "Техполиграфцентр", 2011. - 240 с. - ISBN 978-5-94385-056-1.

[6] Николаев А.Б. Информационные технологии в менеджменте и транспортной логистике: учебное пособие / А.Б. Николаев, А.В. Остроух. – Saint-Louis, MO, USA: Publishing House Science and Innovation Center, 2013. – 254 с. - ISBN 978-0-615-67110-9.

[7] Остроух А.В. Системы искусственного интеллекта в промышленности, робототехнике и транспортном комплексе: монография / А.В. Остроух - Красноярск: Научно-инновационный центр, 2013. – 326 с. - ISBN 978-5-906314-10-9.

[8] Остроух А.В., Николаев А.Б., Збавитель П.Ю., Сальный А.Г. Описание унифицированных программных модулей для лаборатории коллективного пользования // Автоматизация и управление в технических системах. – 2013. – № 2(4). – С. 12-17.