УДК 004.49

ВНЕДРЕНИЕ КОДА В ПРОЦЕСС В ОПЕРАЦИОННОЙ СИСТЕМЕ СЕМЕЙСТВА GNU/LINUX[1]

И. В. Смит

Внедрение программного кода в процесс в операционной системе (ОС) имеет двойное назначение: кроме целей исследовательского характера, оно может преследовать угрозу безопасности компьютерной системы, позволяя нарушителю, например,

— получить доступ к данным, находящимся в виртуальном адресном пространстве этого процесса;
— получить данные в обход криптографических механизмов защиты, реализуемых этим процессом;
— изменять данные процесса.

Но как бы то ни было, знание методов внедрения кода в процесс в ОС несёт в себе несомненный положительный эффект: в одном случае оно расширяет возможности для исследования, в том числе и для обнаружения уязвимостей в ОС, а в другом — мотивирует разработку адекватных методов защиты ОС от соответствующих угроз.

Методы внедрения кода в процессы в ОС Windows, в отличие от ОС семейства Linux, широко известны [1, 2]. В данной работе два таких метода переносятся на ОС GNU/Linux. Условиями для их применения являются:

1) наличие прав доступа пользователя ОС, в процесс которого внедряется код;
2) возможность исполнять системный вызов `ptrace` этим пользователем.

Метод 1 внедрения кода в процесс ОС GNU/Linux

Шаг 1. Открыть процесс на отладку, используя системный вызов `ptrace`.

Шаг 2. Определить адреса библиотечных функций, используемых во внедряемом коде, и разместить их в нем.

Шаг 3. Записать внедряемый код в любую область памяти, имеющую право на выполнение кода процессом.

Шаг 4. Изменить указатель следующей выполняемой команды на адрес внедренного кода.

Шаг 5. Закрыть процесс, используя системный вызов `ptrace`. При этом процесс начинает выполнять внедренный код.

Этот метод имеет следующие сложности реализации:

— внедряемый код должен быть написан на языке ассемблера и быть позиционно независимым, т. е. не зависеть от адреса, по которому его внедрили;
— адреса библиотечных функций, используемых кодом, должны быть вычислены до записи кода;
— необходимые библиотеки могут быть не подгружены в адресное пространство процесса.

Для устранения данных ограничений возможно использование следующего метода.

Метод 2 внедрения кода в процесс ОС GNU/Linux с загрузкой библиотеки

Шаг 1. Написать библиотеку на языке высокого уровня, реализующую необходимые функции.

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).

Шаг 2. Открыть процесс на отладку, используя системный вызов `ptrace`.

Шаг 3. Изменить код библиотеки `ld-linux.so`, находящийся в адресном пространстве процесса, так, чтобы можно было загрузить произвольную библиотеку в адресное пространство процесса.

Шаг 4. Записать код, подгружающий библиотеку из шага 1, в любую область памяти, имеющую право на выполнение кода процессом.

Шаг 5. Изменить указатель следующей выполняемой команды на адрес кода, подгружающего библиотеку.

Шаг 6. Закрыть процесс, используя системный вызов `ptrace`. При этом процесс начинает выполнять код, подгружающий библиотеку.

Эти методы реализованы и опробованы в дистрибутиве `Gentoo`.

ЛИТЕРАТУРА

1. Using Process Infection to Bypass Windows Software Firewalls [Электронный ресурс]. — Режим доступа : `http://www.phrack.org/issues.php?issue=62&id=13`
2. `http://www.phrack.com/issues.html?issue=65&id=10` phook — The PEB Hooker [Электронный ресурс].
3. `http://www.phrack.com/issues.html?issue=59&id=12` Building ptrace injecting shellcodes [Электронный ресурс].

УДК 004.42

РАЗРАБОТКА И РЕАЛИЗАЦИЯ СЕРВЕРА ИГРЫ CTF[1]

Н. О. Ткаченко, Д. В. Чернов

Соревнования по защите информации Capture the Flag (CTF) [1] традиционно проводятся по следующим правилам. Каждой команде перед началом игры выдается одинаковый образ виртуальной машины с какой-либо операционной системой, на которой установлен определенный набор сервисов, содержащих уязвимости. На эти сервисы в процессе игры жюри высылает некоторую информацию, называемую флагами. Цель игры — захватить флаги противника, используя найденные в сервисах уязвимости, и при этом защитить свои флаги, устраняя уязвимости в собственных сервисах. Командам начисляются баллы как за защиту собственного флага, так и за успешное обнаружение чужого. Лучшая и самая простая защита флагов — их удаление, но, по очевидным причинам, такая защита запрещена правилами. Жюри постоянно проверяет сервисы на наличие флагов и снимает очки с команды, если не может получить к ним доступ. В процессе игры командам предоставляется возможность решать дополнительные задания, называемые *квестами*, или давать советы другим командам по увеличению надежности их сервисов, называемые *эдвайзори*. За это командам также начисляются очки.

В классических правилах первый час игры команды изолированы друг от друга межсетевым экраном. Данное время отводится для настройки оборудования и программ, а также для начального ознакомления с сервисами. Затем жюри предоставляет доступ командам к подсетям друг друга и начинает отправлять на сервисы флаги. Во время игры каждая команда имеет доступ к любому сервису противника.

[1]Работа выполнена в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг. (гос. контракт № П1010).