# Организация защиты памяти в процессорах

## Описание

NX (XD) — атрибут страницы памяти в архитектурах x86 и x86-64, который может применяться для более надежной защиты системы от программных ошибок, а также использующих их вирусов, троянских коней и прочих вредоносных программ. NX (No eXecute) — терминология AMD. Intel называет этот атрибут XD-бит (eXecution Disable).

Поскольку в современных компьютерных системах память разделяется на страницы, имеющие определенные атрибуты, разработчики процессоров добавили ещё один: запрет исполнения кода на странице. То есть, такая страница может быть использована для хранения данных, но не программного кода. При попытке передать управление на такую страницу процессор сформирует особый случай ошибки страницы и программа (чаще всего) будет завершена аварийно. Атрибут защиты от исполнения давно присутствовал в других микропроцессорных архитектурах, однако в x86-системах такая защита реализовывалась только на уровне программных сегментов, механизм которых давно не используется современными ОС. Теперь она добавлена ещё и на уровне отдельных страниц.

Современные программы четко разделяют на сегменты кода («text»), данных («data»), неинициализированных данных («bss»), а также динамически распределяемую область памяти, которая подразделяется на кучу («heap») и программный стек («stack»). Если программа написана без ошибок, указатель команд никогда не выйдет за пределы сегментов кода, однако, в результате программных ошибок, управление может быть передано в другие области памяти. При этом процессор перестанет выполнять какие-то запрограммированные действия, а будет выполнять случайную последовательность команд, за которые он будет принимать хранящиеся в этих областях данные, до тех пор, пока не встретит недопустимую последовательность, или попытается выполнить операцию, нарушающую целостность системы, которая вызовет срабатывание системы защиты. В обоих случаях программа завершится аварийно. Также процессор может встретить последовательность, интерпретируемую как команды перехода к уже пройденному адресу. В таком случае процессор войдет в бесконечный цикл, и программа «зависнет», забрав 100 % процессорного времени. Для предотвращения подобных случаев и был введен этот дополнительный атрибут: если некоторая область памяти не предназначена для хранения программного кода, то все её страницы должны помечаться NX-битом, и в случае попытки передать туда управление процессор сформирует особый случай и ОС тут же аварийно завершит программу, сигнализировав выход за пределы сегмента (SIGSEGV).

Основным мотивом введения этого атрибута было не столько обеспечение быстрой реакции на подобные ошибки, сколько то, что очень часто такие ошибки использовались злоумышленниками для несанкционированного доступа к компьютерам, а также написания вирусов. Появилось огромное количество таких вирусов и червей, использующих уязвимости в распространенных программах.

Один из сценариев атак состоит в том, что воспользовавшись переполнением буфера в программе (зачастую это демон, предоставляющий некоторый сетевой сервис), специально написанная вредоносная программа (эксплоит) может записать некоторый код в область данных уязвимой программы таким образом, что в результате ошибки этот код получит управление и выполнит действия, запрограммированные злоумышленником (зачастую это запрос выполнить программу-оболочку ОС, с помощью которой злоумышленник получит контроль над уязвимой системой с правами владельца уязвимой программы, очень часто это root).

## Технические детали

Переполнение буфера часто возникает, когда разработчик программы выделяет некоторую область данных (буфер) фиксированной длины, считая, что этого будет достаточно, но потом, манипулируя данными никак не проверяет выход за её границы. В результате поступающие данные займут области памяти им не предназначенные, уничтожив имеющуюся там информацию. Очень часто временные буферы выделяются внутри процедур (подпрограмм), память для которых выделяется в программном стеке, в котором также хранятся адреса возвратов в вызывающую подпрограмму. Тщательно изучив код программы, злоумышленник может обнаружить такую ошибку, и теперь ему достаточно передать в программу такую последовательность данных, обработав которую программа ошибочно заменит адрес возврата в стеке на адрес, требуемый злоумышленнику, который также передал под видом данных некоторый программный код. После завершения подпрограммы инструкция возврата (RET) вытолкнет из стека в указатель команд адрес входа в процедуру злоумышленника. Контроль над компьютером получен.

Благодаря атрибуту NX такое становится невозможным. Область стека помечается NX-битом и любое выполнение кода в нём запрещено. Теперь же, если передать управление стеку, то сработает защита. Хоть программу и можно заставить аварийно завершиться, но использовать её для выполнения произвольного кода становится очень сложно (для этого потребуется ошибочное снятие программой NX-защиты).

Однако, некоторые программы используют выполнение кода в стеке или куче. Такое решение может быть связано с оптимизацией, динамической компиляцией или просто оригинальным техническим решением. Обычно, операционные системы предоставляют системные вызовы для запроса памяти с разрешенной функцией исполнения как раз для таких целей, однако многие старые программы всегда считают всю память исполнимой. Для запуска таких программ под Windows приходится отключать функцию NX на весь сеанс работы, и чтобы включить её вновь, требуется перезагрузка. Хотя в Windows и предусмотрен механизм белого списка приложений, для которых отключен DEP, тем не менее данный метод не всегда работает корректно.[источник не указан 151 день] Примером такой программы может служить Iris.

NX-бит является самым старшим разрядом элемента 64-битных таблиц страниц, используемых процессором для распределения памяти в адресном пространстве. 64-разрядные таблицы страниц используются операционными системами, работающими в 64-битном режиме, либо с включенным расширением физических адресов (PAE). Если ОС использует 32-разрядные таблицы, то возможности использовать защиту страниц от исполнения нет.