Как Windows раздаёт обновления по сети: полный реверс P2P протокола Windows Update

Posted on 6 апреля, 2026 by AkaTor

Категория: Security Research / Reverse Engineering
Уровень: Advanced (Red Team / Vulnerability Research)
Автор: Aka Tor
Цель: doclient.dll (Delivery Optimization Service, Windows 11 Canary)

Введение

Windows Delivery Optimization (DO/WUDO) — встроенный P2P сервис для распространения обновлений Windows. Работает от **SYSTEM**, слушает на **TCP порту 7680** на всех интерфейсах, позволяет peers в локальной сети обмениваться контентом Windows Update.

Microsoft никогда не публиковал спецификацию этого P2P протокола.

Поверхность атаки

- Порт 7680 открыт на всех интерфейсах (0.0.0.0)
- Контекст SYSTEM (NetworkService)
- Нет аутентификации — только проверка swarm hash
- Нет шифрования на локальном P2P трафике (CDN/cloud — HTTPS)
- Доступен из LAN без учётных данных

---

1. Архитектура сервиса

`dosvc.dll` — тонкий загрузчик, вся логика в `doclient.dll`, загружаемой через

`LoadLibraryExW` → `CreateDOService`.

Ключевые классы (из RTTI):

```
CPeerSock          TCP сокет, state machine, парсинг сообщений
CSwarmConn         Per-peer соединение, handshake, диспатч
CServer            Реестр swarm'ов, управление peers
CSwarm             Состояние download swarm
CDownload          Трекинг скачивания, управление pieces
CStorage           Хранение pieces, disk I/O, hash verification
CBitField          Битовый массив доступности pieces
CMetaInfo          Метаданные контента (кол-во pieces, размер, хэши)
CConnMan           Менеджер соединений, переключение на CDN
CHttpPeerConn      HTTP-mode peer (CDN вместо TCP P2P)
CHash              Обёртка BCrypt (SHA256)
```

---

2. Как работает протокол (обзор)

Протокол имеет три стадии: подключение (handshake со строкой «Swarm protocol» и 32-байтовым хэшем контента), обмен идентификаторами (20 байт peer ID), и передача данных (11 типов сообщений с длиной до 2 МБ). Подробная спецификация — в разделе 12.

Ключевое отличие от BitTorrent: handshake клиента (56 байт) не содержит peer ID, а ответ сервера (76 байт) — содержит. Swarm hash должен совпасть с активным скачиванием, иначе соединение разрывается.

Сервис поддерживает 11 типов сообщений: управление потоком (choke/unchoke/interested), обмен информацией о доступных частях (have/bitfield), запросы и передача данных (request/piece/cancel), и служебные (keepalive, do_not_have).

3. Cloud-инфраструктура

DO использует 5 cloud endpoint’ов для конфигурации, обнаружения peers и получения CDN адресов. Единственный захардкоженный адрес — `geo.prod.do.dsp.mp.microsoft.com`. Все остальные адреса приходят динамически

по цепочке: Geo → GeoVersion → ContentPolicy/KeyValue/Discovery. Все запросы по HTTPS с привязкой сертификата.

Обнаружение peers происходит через DNS-SD (mDNS) в локальной сети, cloud API (ContentPolicy) и Teredo/IPv6.

4. Верификация данных

Каждый piece проверяется SHA256 против таблицы хэшей из объекта `CMetaInfo`. Таблица создаётся **только** из ответа CDN и подписана RSA (PKCS#1). Peers не могут подменить таблицу — она приходит от Microsoft по HTTPS.

Без CDN метаданных P2P скачивание невозможно — `CPieceChecker` не инициализируется и данные от peers не принимаются.

---

5. Особенности безопасности протокола

В ходе тестирования критических уязвимостей не обнаружено. Сервис ни разу не крэшнулся при отправке любых комбинаций сообщений. Отмечены следующие архитектурные особенности:

- **P2P трафик не зашифрован** — TLS negotiation не обнаружен при реверсе. CDN/cloud запросы идут по HTTPS
- **Аутентификация peers** — только через совпадение swarm hash с активным download. Без hash подключение отклоняется
- **Целостность данных** — каждый piece проверяется SHA256 против таблицы хэшей полученной от Microsoft CDN. Подмена данных невозможна
- **Устойчивость сервиса** — при массовой отправке сообщений (до 100 подряд) процесс сервиса выживает, закрываются только отдельные соединения
- **Цепочка доверия CDN** — адреса обновляются по HTTPS с привязкой сертификата. Таблица хэшей подписана RSA

---

6. HTTP Mode

Альтернатива TCP P2P. `CHttpPeerConn` использует HTTP Range requests через WinHTTP:

```
// CHttpPeerConn::SendRequestBlock — декомпиляция
// Offset вычисляется: GetPieceOffset(piece_index) + request.Begin()
// Запрос через WinHTTP с Range header

// Обработка ответов:
// 206 → Content-Range парсинг (bytes start-end/total)
// 200 → Content-Length или Transfer-Encoding: chunked
// 403 → CDN unauthorized, banning логика
// 301/302 → redirect с лимитом

// Error handling:
// Fibonacci backoff для retries (CFibonacciBackoff)
// CDN ban list (CBanList::IsBanned)
// Переключение на альтернативный CDN (_TryCdnFallbackDownload)
```

---

7. Режимы скачивания

Mode	Название	Описание
0	HTTP only	Без P2P
1	LAN P2P + HTTP	По умолчанию
2	Group P2P + HTTP	Enterprise
3	Internet P2P + HTTP	P2P через интернет
99	Simple HTTP	CDN provider

Конфигурация: реестр `HKLM\SOFTWARE\...\DeliveryOptimization` или cloud override через GeoVersion/KeyValue endpoints.

8. Vtable CSwarmConnPeerSockListener

Полная vtable из 22 методов обратного вызова (из IDA RTTI):

Offset	Метод	Описание
0x00	GetConnMan	Получить менеджер соединений
0x08	GetMaxRequests	Макс. количество запросов
0x10	GetPeerType	Тип peer
0x18	IsUnrestrictedLanConn	Неограниченное LAN соединение?
0x20	ScheduleDeletion	Запланировать удаление
0x28	OnConnect	Подключение установлено
0x30	OnInvalidMsg	Невалидное сообщение
0x38	OnKeepAlive	Keepalive получен
0x40	OnPartialHandshake	Handshake получен
0x48	OnHandover	Передача управления
0x50	OnPeerId	Peer ID получен
0x58	OnChoke	Choke получен
0x60	OnUnchoke	Unchoke получен
0x68	OnInterested	Interested получен
0x70	OnNotInterested	NotInterested получен
0x78	OnHave	Piece доступен
0x80	OnDoNotHave	Piece недоступен

0x88	OnBitField	Bitfield получен
0x90	OnRequest	Запрос piece
0x98	OnPiece	Piece data получен
0xA0	OnCancel	Отмена запроса
0xA8	OnBytesReceived	Статистика приёма

---

9. Инструменты тестирования

В ходе исследования разработаны инструменты для тестирования протокола. Все работают от обычного пользователя — привилегии администратора не требуются.

11.1 Формирование handshake

```
// Формирование P2P handshake для DoSvc
void send_handshake(SOCKET s, const char* swarm_hash_hex) {
    unsigned char buf[128];
    int pos = 0;

    buf[pos++] = 14;                              // proto_len
    memcpy(buf + pos, "Swarm protocol", 14);       // protocol string
    pos += 14;
    buf[pos++] = 1;                               // version
    memset(buf + pos, 0, 8);                      // reserved
    pos += 8;

    unsigned char hash[32] = {0};
    hex2bytes(swarm_hash_hex, hash, 32);          // swarm hash (32
bytes)
    memcpy(buf + pos, hash, 32);
    pos += 32;

    send(s, (char*)buf, pos, 0);                  // total: 56 bytes
}
```

11.2 Отправка P2P сообщений

```
// Big-endian uint32
void put32(unsigned char* buf, unsigned int val) {
    buf[0] = (val >> 24) & 0xFF;
    buf[1] = (val >> 16) & 0xFF;
    buf[2] = (val >> 8) & 0xFF;
    buf[3] = val & 0xFF;
}

// Отправка length-prefixed сообщения
void send_msg(SOCKET s, unsigned char* msg, int msglen) {
    unsigned char hdr[4];
    put32(hdr, msglen);          // length prefix (4 bytes BE)
    send(s, (char*)hdr, 4, 0);
    send(s, (char*)msg, msglen, 0);
}

// Пример: отправка HAVE (msgId=4, piece_index=42)
unsigned char msg[5];
msg[0] = 4;                     // msgId = HAVE
put32(msg + 1, 42);             // piece_index = 42
send_msg(s, msg, 5);

// Пример: отправка REQUEST (msgId=6)
unsigned char req[13];
req[0] = 6;                     // msgId = REQUEST
put32(req + 1, 0);              // piece_index
put32(req + 5, 0);              // offset
put32(req + 9, 1048576);        // length (1MB)
send_msg(s, req, 13);
```

11.3 Результаты теста массовой отправки

Отправка повторяющихся сообщений на одном соединении показала что сервис закрывает соединения при определённых комбинациях:

```
msgId=0 (Choke):          соединение закрыто на повторе 2
msgId=1 (Unchoke):        50 повторов — выжил
```

```
msgId=4 (Have):           закрыто на повторе 46
msgId=5 (Bitfield):       закрыто на повторе 4
msgId=6 (Request):        закрыто на повторе 5
msgId=7 (Piece):          закрыто на повторе 6
msgId=8 (Cancel):         закрыто на повторе 5
msgId=9:                  50 повторов — выжил
msgId=10 (DoNotHave):     закрыто на повторе 12

Процесс сервиса (svchost/DoSvc) не крашится — закрываются
только индивидуальные соединения. Сервис остаётся доступен.
```

---

10. Полная спецификация протокола

10.1 Обзор

BitTorrent-подобный P2P протокол для распространения контента Windows Update. Бинарный фрейминг, big-endian целые числа, TCP транспорт. Порт 7680.

10.2 Handshake (State 0)

Направление	Поле	Размер	Описание
Client → Server (56 байт)	proto_len	1	Длина protocol string (14)
	protocol	14	«Swarm protocol» (ASCII, case sensitive)
	version	1	Версия протокола (1)
	reserved	8	Нули
	swarm_hash	32	SHA256 идентификатор контента

Server → Client (76 байт)	proto_len	1	14
	protocol	14	«Swarm protocol»
	version	1	min(local, remote)
	reserved	8	reserved[5] = 0x10
	swarm_hash	32	Echo
	peer_id	20	Peer ID сервера

Валидация:

- `proto_len`: диапазон 1-49, не может быть 0 или 22
- `protocol`: точное совпадение «Swarm protocol»
- `swarm_hash`: lookup в `CServer::GetSwarm` — должен совпадать с активным download

10.3 Peer ID (State 1)

Поле	Размер	Описание
peer_id	20 байт	Произвольный идентификатор peer’a

10.4 Фрейминг сообщений (State 2)

Поле	Размер	Описание
length	4 (BE)	Размер payload, макс 0x1F4000 (2 MB). 0 = keepalive
msgId	1	Тип сообщения
payload	length-1	Данные сообщения

10.5 Все типы сообщений

msgId	Название	Payload	Wire формат
0	CHOKE	0	`00000001 00`
1	UNCHOKE	0	`00000001 01`

2	INTERESTED	0	`00000001 02`
3	NOT_INTERESTED	0	`00000001 03`
4	HAVE	4	`00000005 04 [index:4BE]`
5	BITFIELD	N	`[N+1:4BE] 05 [bitfield:N]`
6	REQUEST	12	`0000000D 06 [idx:4BE][off:4BE][len:4BE]`
7	PIECE	8+N	`[N+9:4BE] 07 [idx:4BE][off:4BE][data:N]`
8	CANCEL	12	`0000000D 08 [idx:4BE][off:4BE][len:4BE]`
10	DO_NOT_HAVE	4	`00000005 0A [index:4BE]` (условный)
20	KEEPALIVE_DATA	N	`[N+1:4BE] 14 [data:N]` (отбрасывается)

BE = big-endian. Все msgId вне таблицы → `OnInvalidMsg` → ошибка.

10.6 Cloud Endpoints

Endpoint	URL	Назначение
Geo	https://geo.prod.do.dsp.mp.microsoft.com/geo	Начальная точка (захардкожен), геолокация
GeoVersion	https://geover.prod.do.dsp.mp.microsoft.com/geoversion	Версия конфигурации, обновление endpoint’ов
ContentPolicy	https://cp801.prod.do.dsp.mp.microsoft.com/v3/content	Peer discovery, CDN URL’ы для контента
KeyValue	https://kv801.prod.do.dsp.mp.microsoft.com/	Удалённая конфигурация (ключ-значение)
Discovery	https://disc801.prod.do.dsp.mp.microsoft.com/	Обнаружение peers для swarm

Цепочка обновления: Geo (захардкожен) → GeoVersion → обновляет адреса ContentPolicy, KeyValue, Discovery. Все запросы по HTTPS.

10.7 Peer Discovery

Метод	Описание	Конфигурация
DNS-SD (mDNS)	Обнаружение в LAN	DnsPeerDiscoveryParticipationRate, DnsPeerSearchIntervalMsecs
Cloud API	Запрос ContentPolicy с swarm ID	ContentPolicy_EndpointFullUri
Teredo/IPv6	Через TeredoExtAcquireTeredoConsumerHandle	—

10.8 Верификация данных

Что	Алгоритм	Размер	Источник доверия
Piece hash	SHA256 (BCrypt)	32 байта	CMetaInfo из CDN (HTTPS)
Swarm hash	SHA256	32 байта	Вычисляется из метаданных
Content signature	RSA + SHA256 (PKCS#1)	Переменный	Сертификат Microsoft

Цепочка доверия: Microsoft CDN (HTTPS + RSA подпись) → метаданные + таблица SHA256 хэшей → CMetaInfo. Peers отправляют данные → SHA256(data) → сравнение с CMetaInfo.hash_table[piece_index]. Без CDN метаданных P2P download невозможен — peers не могут подменить таблицу хэшей.

10.9 Транспортная безопасность

Канал	Шифрование	Аутентификация
P2P (TCP 7680)	Нет (открытый текст)	Только swarm hash
CDN (HTTP/HTTPS)	TLS через WinHTTP	Привязка сертификата (CHttpServerCertRetriever)
Cloud API	HTTPS	TLS + RSA подпись метаданных

10.10 Режимы скачивания

Mode	Название	P2P
0	HTTP Only	Нет
1	LAN P2P + HTTP	Только LAN (по умолчанию)
2	Group P2P + HTTP	Enterprise группа
3	Internet P2P + HTTP	Через интернет
99	Simple HTTP	Нет (CDN provider)

10.11 Конфигурация (реестр + cloud override)

Путь:
`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\DeliveryOptimization`

Ключ	Описание
DODownloadMode	Режим скачивания (0-3, 99)
DOAbsoluteMaxCacheSize	Максимальный размер кэша
DOAllowVPNPeerCaching	Разрешить P2P через VPN
DOMinBackgroundQoS	Минимальный фоновый QoS
DOCacheHost	Hostname кэш-сервера
DODelayBackgroundDownloadFromHttp	Задержка перед переключением на HTTP (фон)
DODelayForegroundDownloadFromHttp	Задержка перед переключением на HTTP (передний план)
ParticipationRate	Уровень участия в P2P
RestrictPeerSelectionBy	Ограничение выбора peers
VpnKeywords	Ключевые слова для определения VPN адаптеров

10.12 Полная Vtable CSwarmConnPeerSockListener

Offset	Метод	Описание
0x00	GetConnMan	Менеджер соединений
0x08	GetMaxRequests	Макс. запросов
0x10	GetPeerType	Тип peer
0x18	IsUnrestrictedLanConn	Неограниченный LAN?
0x20	ScheduleDeletion	Удаление
0x28	OnConnect	Соединение
0x30	OnInvalidMsg	Невалидное сообщение
0x38	OnKeepAlive	Keepalive
0x40	OnPartialHandshake	Handshake
0x48	OnHandover	Передача
0x50	OnPeerId	Peer ID
0x58	OnChoke	Choke
0x60	OnUnchoke	Unchoke
0x68	OnInterested	Interested
0x70	OnNotInterested	NotInterested
0x78	OnHave	Piece доступен
0x80	OnDoNotHave	Piece недоступен
0x88	OnBitField	Bitfield
0x90	OnRequest	Запрос piece
0x98	OnPiece	Piece data
0xA0	OnCancel	Отмена
0xA8	OnBytesReceived	Статистика приёма

0xB0	OnBodyBytes	Байты тела
0xB8	OnBytesSent	Отправлено
0xC0	OnCommError	Ошибка коммуникации

---

11. Методология

1. **Service Discovery:** `netstat` → порт 7680 → DoSvc → doclient.dll
2. **Статический анализ:** IDA Pro + MCP — exports, imports, строки
3. **State Machine:** декомпиляция `CPeerSock::_ProcessBytes` → 3 состояния
4. **Message Parser:** декомпиляция `CPeerSock::_ProcessMsg` → 11 типов
5. **Vtable:** RTTI `CSwarmConnPeerSockListener` → 22 метода обратного вызова
6. **Send Functions:** декомпиляция SendHandShake, SendHave, SendBitField, SendRequest
7. **Cloud Endpoints:** строковые ссылки → 5 cloud service URL'ов
8. **Hash Verification:** `CPieceChecker::CheckPiece` → BCrypt SHA256
9. **Конфигурация:** таблица строк → 30+ ключей конфигурации
10. **Тестирование:** кастомные C-инструменты → анализ поведения протокола

---

12. Рекомендации по защите

- **Отключить P2P:** DODownloadMode=0 через Group Policy или реестр
- **Только LAN:** оставить Mode 1 (по умолчанию), не включать Mode 3 (internet P2P)
- **Firewall:** блокировать входящий TCP 7680 из недоверенных сегментов сети
- **Мониторинг:** следить за аномальными подключениями к порту 7680 (IDS/EDR)
- **VLAN:** ограничить P2P трафик доверенными сегментами сети

Тестирование проводилось на собственном оборудовании в изолированной лабораторной среде. Используйте полученные знания ответственно и только с разрешения владельца системы.