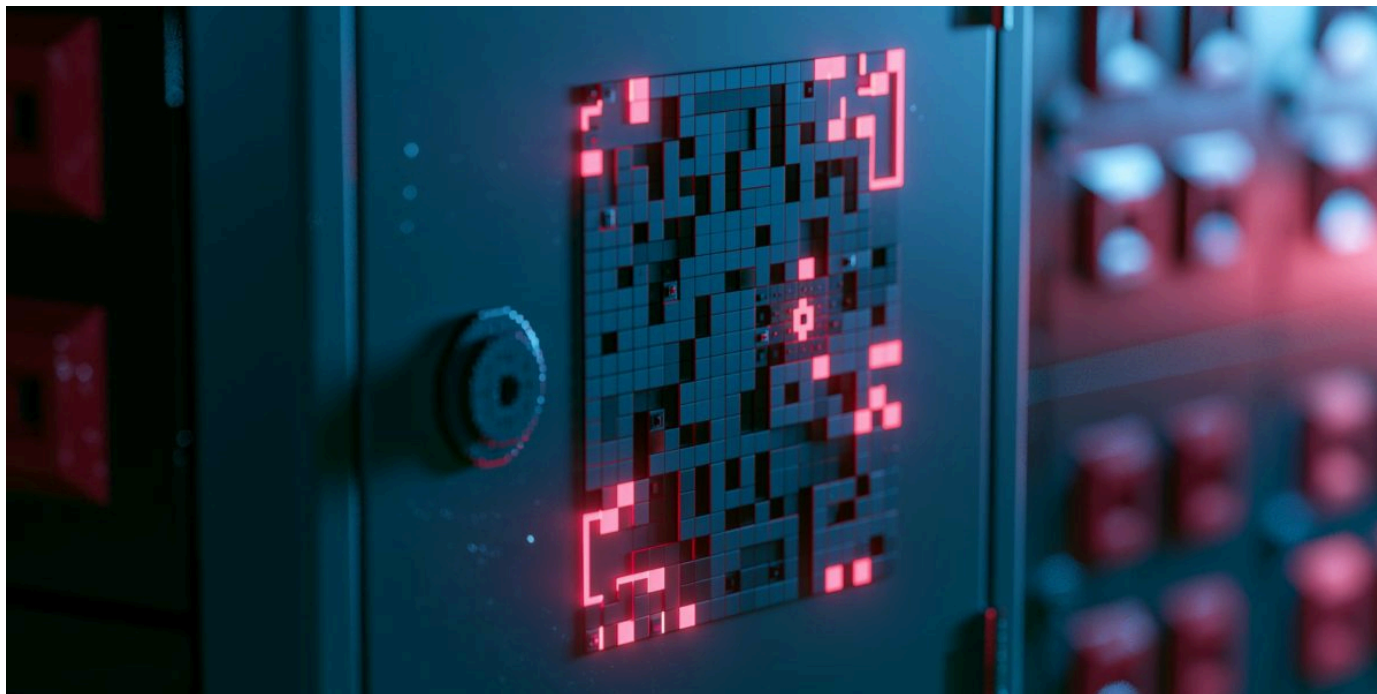


Новый шифровальщик GenieLocker для Windows, ESXi и Linux

SL securelist.ru/genielocker-ransomware-for-windows-linux-and-esxi/116468

Федор Синецын, Янис Зинченко

July 30, 2026



Авторы

Введение

Новое семейство шифровальщиков GenieLocker, активное с марта 2026 года, было замечено в атаках на российские организации преимущественно из промышленного сектора. На основании [данных из открытых источников](#) эти атаки были атрибутированы группе Toy Ghoul.

Группа Toy Ghoul, также известная как Bearlyfy, Labubu и Laboo.bo0, специализируется на финансово мотивированном вымогательстве. Ранее она использовала сторонние шифровальщики, включая RedAlert, LockBit и Babuk. По всей видимости, GenieLocker является собственной разработкой группы, призванной расширить ее инструментарий и снизить зависимость от стороннего вредоносного ПО. Нам удалось обнаружить множество образцов GenieLocker двух разновидностей: PE-сборки для Windows и ELF-сборки для Linux и ESXi.

Технические детали

Схема атаки

Типичные тактики, техники и процедуры группы Toy Ghoul мы уже рассматривали в [предыдущей публикации](#).

В этой статье мы подробно разберем возможности кастомизированного троянца-шифровальщика GenieLocker для Windows и Linux. В качестве примера мы приведем краткий обзор атаки, произошедшей в конце марта 2026 года, в ходе которой GenieLocker был развернут в инфраструктуре жертвы.

Первоначальный доступ

В ходе инцидента злоумышленники первоначально проникли в инфраструктуру через соединение OpenVPN из сети внешнего партнера. Предположительно, они воспользовались доверительными отношениями с этой организацией и подключились с помощью украденных учетных данных, которые на тот момент были все еще действительны.

Разведка и доступ к учетным данным

Проникнув в целевую сеть, злоумышленники установили на скомпрометированные хосты дополнительные инструменты, включая OpenSSH, socks5.exe, SoftPerfect Network Scanner и Mimikatz. Для проведения разведки они использовали SoftPerfect Network Scanner, а для извлечения учетных данных — Mimikatz. Криминалистический анализ также показал, что злоумышленники получили доступ к

менеджеру паролей KeePassXC, который был установлен в нескольких скомпрометированных системах. По всей видимости, атакующие пытались извлечь сохраненные логины и пароли из баз данных KeePass.

Горизонтальное перемещение и C2-коммуникация

Горизонтальное перемещение в среде Windows осуществлялось с помощью RDP, а доступ к Linux-серверам злоумышленники получали через SSH. Для массового развертывания шифровальщика злоумышленники использовали легитимные утилиты PsExec и PAExec. Кроме того, они настроили обратный SSH-туннель для поддержания связи с командным сервером.

Воздействие

На следующем этапе атаки злоумышленники зашифровали файлы на скомпрометированных компьютерах на базе Windows с помощью PE-версии программы-вымогателя GenieLocker. На серверах Linux и ESXi они остановили работающие виртуальные машины и зашифровали их диски с помощью ELF-версии GenieLocker.

Наблюдаемые тактики, техники и процедуры соответствуют тем, которые ранее были зафиксированы в атаках, приписываемых группе Toy Ghoul. Как и в предыдущих инцидентах, в ходе криминалистического анализа не было обнаружено признаков эксфильтрации данных. Это характерная особенность операций группы: Toy Ghoul не ведет сайты для публикации украденных данных и в описываемом случае тоже не использовала модель двойного вымогательства.

Шифровальщик для Windows

Версия GenieLocker для Windows (MD5: 5d62c1349b8981c396c9a23f4f8f053c) преимущественно написана на языке C, но скомпилирована с использованием библиотек C++ с помощью Microsoft Visual C/C++. Этот зловред включает ряд функций, типичных для программ-вымогателей, среди которых: завершение процессов, остановка служб, обход отладчиков и использование усложненной схемы шифрования. Для реализации криптографических операций используется библиотека с открытым исходным кодом libsodium.

В соответствии с недавней тенденцией, подтвержденной нашими наблюдениями за атаками с применением других семейств программ-вымогателей, GenieLocker не сохраняет файлы с требованием выкупа в системе жертвы. Троянец не содержит контактных данных злоумышленников или адресов для ведения переговоров. Вместо этого атакующие загружают требования о выкупе и контактную информацию вручную в ходе атаки. Вероятно, таким образом разработчики GenieLocker стремятся избежать проактивного обнаружения шифровальщика при выявлении массового создания файлов-записок.

```
" +-----+\n"
" | GenieLocker v%d | \n"
" +-----+\n"
"\n"
" Usage: genie_encrypt [options] [path]\n"
"\n"
" If no path given, encrypts all local + remote drives.\n"
"\n"
" Options:\n"
" -p, --percent N      Encryption percentage (0-100, default %d)\n"
"                      0 = encrypt only head (%s)\n"
" -r, --recursive      Process directories recursively\n"
" -l, --log FILE        Log file path\n"
" -h, --help           Show this help\n"
"\n"
" Build config:\n"
" Extensions:  %s\n"
" Vault ext:   %s\n"
" Head encrypt: %s\n"
" Start delay: %d sec\n"
" Self-delete: %s
```

Аргументы и запуск

GenieLocker поддерживает несколько аргументов, определяющих его поведение.

Аргумент	Описание
Первый аргумент	«Секретный аргумент», строковое шестнадцатеричное значение
-p, —percent N	Процент шифруемого содержимого файла
-r, —recursive	Рекурсивная обработка каталогов
-l, —log <имя_файла>	Задание пути к лог-файлу
-h, —help	Вывод справочного сообщения
Последний аргумент	Путь к шифруемым файлам

В качестве первого аргумента GenieLocker ожидает шестнадцатеричную строку, которая в коде зловреда обозначена как «секретный аргумент». Скорее всего, этот механизм должен предотвращать выполнение программы-вымогателя в песочницах и других средах автоматизированного анализа. Не исключено также, что таким образом разработчики ограничивают возможность несанкционированного использования шифровальщика другими злоумышленниками.

```
SetConsoleOutputCP(65001u);
SetConsoleCP(65001u);
if ( _argc < 2 )
{
    stderr = _acrt_iob_func(2u);
    Log(stderr, " Missing secret argument\n");
    return 1;
}
```

Проверка секретного аргумента

Секретный аргумент представляет собой шестнадцатеричную строку переменной длины, не превышающей 4096 байт. Она преобразуется в массив байтов, для которого вычисляется хэш SHA-256. Результат сравнивается со значением, заданным непосредственно в коде. Если значения совпадают, к секретному аргументу добавляется фиксированная строка `session`, после чего итоговая строка хэшируется с помощью алгоритма BLAKE2b-256. Однако полученный хэш в дальнейшем нигде не используется. Возможно, он нужен для функциональности, которая все еще разрабатывается.

```
// BLAKE2b-256
crypto_generichash_init(&s->blake_ctx, 0i64, 0i64, 32i64);
crypto_generichash_update(&s->blake_ctx, hex_secret, hex_secret_size);
crypto_generichash_update(&s->blake_ctx, "session", 7ui64);
crypto_generichash_final(&s->blake_ctx, &state->secret_blake2b, 0x20u);
```

Хэширование секретного значения

Защита от отладки

GenieLocker проверяет наличие отладки своего процесса несколькими методами. После запуска шифровальщик сначала выполняет проверку среды (в оригинальном коде `Environment check`), используя функции WinAPI `IsDebuggerPresent` и `CheckRemoteDebuggerPresent` для обнаружения отладчика.

```

if ( IsDebuggerPresent()
    || (pbDebuggerPresent = 0,
        CurrentProcess = GetCurrentProcess(),
        CheckRemoteDebuggerPresent(CurrentProcess, &pbDebuggerPresent),
        pbDebuggerPresent) )
{
    stderr = _acrt_iob_func(2u);
    LOG(stderr, " Environment check failed\n");
    return 0i64;
}

```

Проверка среды

После проверки секретного аргумента GenieLocker создает параллельный поток с именем watchdog. Он работает в бесконечном цикле и каждые 500 миллисекунд выполняет ряд довольно известных проверок для обнаружения отладочных средств. Если хотя бы одну из проверок пройти не удастся, весь процесс GenieLocker немедленно завершается.

```

if ( IsDebuggerPresent() )
    CloseProgram(s);
// 0x70 - FLG_HEAP_ENABLE_TAIL_CHECK |
//         FLG_HEAP_ENABLE_FREE_CHECK |
//         FLG_HEAP_VALIDATE_PARAMETERS
if ( (NtCurrentPeb()->NtGlobalFlag & 0x70) != 0 )
    CloseProgram(s);
ProcessHeap = NtCurrentPeb()->ProcessHeap;
if ( ProcessHeap->Flags != 2 || ProcessHeap->ForceFlags )
    CloseProgram(s);
pbDebuggerPresent = 0;
CurrentProcess = GetCurrentProcess();
CheckRemoteDebuggerPresent(CurrentProcess, &pbDebuggerPresent);
if ( pbDebuggerPresent )
    CloseProgram(s);
if ( CheckHardwareBreakpoints() )
    CloseProgram(s);
if ( CheckVectoredExceptions() )
    CloseProgram(s);
if ( CheckWindowsWithDebuggerNames() )
    CloseProgram(s);
if ( SearchDebuggerProcessNames() )
    CloseProgram(s);
if ( TextSectionCheckCRC32(&crc32) )
    CloseProgram(s);
Sleep(500u);

```

Проверки в потоке watchdog

Отдельно стоит отметить, что при запуске потока watchdog процесс GenieLocker вычисляет хэш CRC32 для своей секции .text и сохраняет полученное значение. Затем на каждом цикле выполнения он повторно рассчитывает хэш и сравнивает его с исходным. Таким образом троянец может обнаружить модификацию кода секции, если он был изменен отладчиком или другой программой.

Подготовка к шифрованию

GenieLocker содержит несколько списков исключений. Например, он не шифрует каталоги с именами из приведенного ниже списка. В него преимущественно входят системные папки, шифрование которых приведет к повреждению операционной системы.

```
1 $recycle.bin;config.msi;$windows.~bt;$windows.~ws;windows;boot;program files;program files (x86);programdata;system volume information;tor browser;windows.old;intel;msocache;perflogs;x64dbg;public;all users;default;microsoft;appdata
```

Не шифруются и следующие системные файлы Windows:

```
1 autorun.inf;boot.ini;bootfont.bin;bootsect.bak;desktop.ini;iconcache.db;ntldr;ntuser.dat;ntuser.dat.log;ntuser.ini;thumbs.db;GDIPFONTCACHEV1.DA
```

Также исключены из шифрования перечисленные ниже расширения файлов.

```
1 386;adv;ani;bat;bin;cab;cmd;com;cpl;cur;deskthemepack;diagcab;diagcfg;diagpkg;dll;drv;exe;hlp;icl;icns;ico;ics;idx;ldf;lnk;mod;mpa;msc;msh;mssty  
ms;MD
```

Кроме того, троянец содержит список исключений для имен хостов. Зловред определяет имя компьютера с помощью функции GetComputerNameA и сравнивает его со значениями из этого списка. Однако в проанализированном образце список исключений пуст.

```
" +-----+ \n"  
" | HOST WHITELISTED - ABORTING | \n"  
" | Hostname: %-36s | \n"  
" +-----+ \n"  
Текст, выводимый при запуске на хосте из списка исключений
```

Если имя хоста отсутствует в списке исключений, GenieLocker начинает принудительно завершать процессы, которые могут использовать целевые файлы и препятствовать их шифрованию. Перечень таких процессов приведен ниже. Для их остановки троянец использует функцию TerminateProcess.

```
1 sql;oracle;ocssd;dbsnmp;synctime;agntsvc;isqlplussvc;xfssvcon;mydesktopservice;ocautoupds;encsvc;firefox;tbirdconfig;mydesktopqos;ocomm;dl
```

Кроме того, троянец останавливает следующие службы с помощью функции ControlService, передавая код управления SERVICE_CONTROL_STOP:

```
1 vss;sql;svc$;memtas;mepocs;msexchange;sophos;veeam;backup;GxVss;GxBlr;GxFWD;GxCVD;GxCIMgr;1c;Mssqlserver;vmwp;vmms;vmcompute
```

Наконец, GenieLocker запускает потоки шифрования и пытается найти все доступные диски, в том числе сетевые папки, для последующего шифрования.

```
" +-----+ \n"  
" | GenieLocker v%d | \n"  
" +-----+ \n"  
" | Threads: %-4d Head: %-10s Pct: %3d%% | \n"  
" | Extension: %-33s | \n"  
" +-----+ \n"  
Вывод информации о потоках
```

Шифрование файлов и криптография

Расширение зашифрованных файлов зашито непосредственно в код троянца. В описываемом образце это .03ffc1c4a3da0f02. Перед началом шифрования каждого файла GenieLocker создает два вспомогательных файла:

- lock-файл вида <имя_файла.расширение_файла>.03ffc1c4a3da0f02.lock;
- файл-журнал вида <имя_файла.расширение_файла>.03ffc1c4a3da0f02.journal.

Lock-файл предназначен для защиты файлов от повторного шифрования другими потоками или экземплярами зловреда. В этом файле троянец сохраняет текущий PID, полученный с помощью функции GetCurrentProcessId.

Файл журнала содержит жестко заданную строку VCJOURN, значение 1 (вероятно, номер версии), несколько неиспользуемых полей с нулевыми значениями, общее количество блоков, подлежащих шифрованию, количество фактически зашифрованных блоков и контрольную сумму CRC32, которая нужна для проверки целостности содержимого файла.

```

; GLJournal
db 'V', 'C', 'J', 'O', 'U', 'R', 'N', 0 ; magic
db 1 ; version
dq 0 ; field_0
dq 0 ; field_1
dq 0 ; field_2
dq 0 ; field_3
dq 20h ; total_blocks
dq 20h ; encrypted_blocks
dd 748B9E8Dh ; crc32

```

Содержимое файла журнала

По умолчанию GenieLocker шифрует файлы блоками размером 0x1000000 байт. Если передан аргумент -p, определяющий процент шифруемого содержимого файла, вымогатель рассчитывает количество блоков размером 0x1000000 байт, необходимых для покрытия указанной части данных. Каждый блок выбирается в случайном месте внутри файла. Первый блок в начале файла шифруется всегда, независимо от того, задан ли процент шифрования или его значение равно нулю.

Для шифрования содержимого файлов троянец использует алгоритм аутентифицированного шифрования с присоединенными данными (AEAD) XChaCha20-Poly1305 с уникальным ключом и одноразовым числом (nonce) для каждого файла. Кроме того, GenieLocker добавляет в конец файла служебный блок, содержащий метаданные и данные, необходимые для последующей расшифровки. Блок метаданных шифруется с использованием того же алгоритма и ключа, что и содержимое файла, но с другим одноразовым числом. Ключ файла шифруется по схеме Curve25519-XSalsa20-Poly1305 с использованием открытого мастер-ключа злоумышленников, зашитого в код троянца.

Метаданные каждого зашифрованного файла содержат следующие поля:

Значение или имя	Размер в байтах	Описание
version	1	Жестко заданный байт со значением 1, вероятнее всего, версия.
encryption_percent	1	Процент шифруемого содержимого файлов, переданный через аргумент -p.
file_nonce	24	Одноразовое число, использованное при шифровании содержимого файла.
original_filesize	8	Исходный размер файла до шифрования.
total_chunk_count	8	Максимальное количество блоков, которое может содержаться в текущем файле.
chunk_size	4	Размер одного зашифрованного блока (по умолчанию 0x1000000 байт для Windows и 0x400000 байт для ESXi и Linux).
remain_size	4	Количество байтов, оставшихся после разделения содержимого файла на блоки.
blake2b_digest_of_chunks	32	Хэш BLAKE2b-256, вычисленный на основе исходных данных всех блоков до их шифрования. Используется для проверки целостности данных.
chunk_count	4	Количество зашифрованных блоков.
extension	64	Строка с дополнительным расширением шифровальщика.
poly1305_tags (массив)	16 байт на блок	Массив тегов Poly1305 зашифрованных блоков.
bitmask	Размер варьируется, один бит на каждый блок	Битовая маска блоков: заданные биты соответствуют зашифрованным блокам.

Битовая маска содержит количество битов, равное максимальному числу блоков, на которые может быть разбит файл при полном (100%) шифровании. Если бит с определенным индексом равен 1, соответствующий блок файла зашифрован. Если бит равен 0, блок не

зашифрован. Поскольку троянец шифрует файлы в соответствии с заданным процентом, он должен фиксировать, какие именно блоки были зашифрованы.

```
db 1 ; version
db 3Dh ; encryption_percent
db 2Ah, 0A2h, 0E1h, 1Ah, 19h, 54h, 0B8h, 0ADh, 0FBh, 1, 0BCh, 3, 98h, 42h; file_nonce
db 2Dh, 18h, 9Ah, 8Ah, 74h, 0E1h, 0F2h, 1Dh, 65h, 0D1h; file_nonce
dq 7E12DBCh ; original_filesize
dq 20h ; total_chunk_count
dd 400000h ; chunk_size
dd 212DBCh ; remain_size
db 15h, 80h, 9Fh, 5Ah, 7Fh, 74h, 0EAh, 95h, 50h, 70h, 0CEh, 15h, 0E8h; blake2b_digest_of_chunks
db 85h, 0B0h, 0ECh, 0D2h, 0D6h, 16h, 89h, 0F9h, 2Eh, 99h, 0A2h, 53h, 0E7h; blake2b_digest_of_chunks
db 98h, 66h, 18h, 0F4h, 0CBh, 0DEh ; blake2b_digest_of_chunks
dd 18h ; chunk_count
db 2Eh, 34h, 38h, 34h, 33h, 63h, 35h, 61h, 63h, 61h, 39h, 66h, 39h, 33h; extension
db 31h, 38h, 32h, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; extension
db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; extension
db 0, 0, 0, 0, 0, 0 ; extension
```

Структура метаданных в конце зашифрованного файла (без массива тегов Poly1305 и битовой маски)

Шифровальщик для ESXi и Linux

По сравнению с версией для Windows вариант GenieLocker для Linux и ESXi (MD5: 9201e35e2993612612919a3c71302cab) устроен проще: в нем отсутствуют секретный аргумент, механизмы поиска отладчиков и списки исключений. В то же время образец содержит функциональность, специфичную для ESXi, включая поддержку запуска в режиме демона посредством двойного вызова `fork()` и возможность изменения текста приветственного сообщения. В образце указана версия v1, и он, как и Windows-версия, использует криптографическую библиотеку `libsodium`.

```
GENIELOCKER v1 ESXi/Linux x64

GLOBAL OPTIONS:
  -h                Display help information

PROCESSING OPTIONS:
  -p <perc>         Encryption percentage (0-100, default 2, 0: Encrypt only file header)
  -j <workers>      Parallel worker threads (0: Auto-detect CPU cores)
  -r <dir>          Recursive directory processing (default: /vmfs/volumes)

RUNTIME CONTROL:
  -w <sec>          Initial delay before execution
  -d                Daemon mode (background process)
  -l <logfile>      Log output destination

BUILD CONFIGURATION:
  Target extensions:
  Lock extension:    .4843c5aca9f93182
  Auto-start delay:  0 seconds
  Self-destruct:     no

USAGE:
  genie_encrypt [options] <target_path>
```

Описание версии для ESXi

Вывод справки в командной строке оформлен в стиле LockBit, что подтверждает предположение о том, что разработчики GenieLocker изначально стремились создать похожую на LockBit альтернативу для проведения собственных операций.

```
LOCKBIT5.0 ChuongDong Locker v1.01 Windows x64

USAGE
chuongdong64.exe [options]
* Command line length is limited to 500 characters.

BASIC OPTIONS
-h Show this help
-p <dirs> Semicolon-separated list of directories to encrypt
-b <dirs> Semicolon-separated list of directories to bypass

OPERATION MODES
-i Invisible mode (don't change extensions, no notes, don't change modification date)
-v Run in verbose visible mode with status bar in console
  * Not available when using -p
-d Run in visible mode with debug output

NOTES SETTINGS
-n <0/1/2> Notes storage mode (0: none, 1: everywhere, 2: C:\ only)
  * This option is ignored when using -i (invisible mode)
```

Экран справки LockBit, вероятно, послуживший основой для оформления ESXi-версии GenieLocker

На основании пути к каталогу шифрования по умолчанию (/vmfs/volumes) можно сделать вывод, что данная версия ориентирована прежде всего на платформу ESXi. Тем не менее образец также запускается в среде Linux.

Аргумент	Описание
-p <процент>	Процент шифруемого содержимого файла
-j <число_потоков>	Количество потоков шифрования
-r <каталог>	Рекурсивная обработка каталогов
-w <число_секунд>	Задержка перед запуском
-d	Перевод процесса в режим демона
-l <лог-файл>	Путь к лог-файлу

Особенности на ESXi и Linux

Эта сборка поддерживает перевод процесса в режим демона с помощью флага -d, реализованный через классический метод двойного вызова `fork()`, благодаря которому дочерний процесс полностью отделяется от родительского.

Кроме того, образец изменяет файл `/etc/vmware/welcome`, в котором хранится приветственное сообщение (Message of the Day, MOTD) в операционной системе ESXi. В дистрибутивах Linux эта функциональность не работает, поскольку там для хранения MOTD применяются другие пути. В исследованном образце GenieLocker содержимое данного сообщения было пустым.

Версия для ESXi также поддерживает ряд базовых возможностей, отсутствующих в разновидности для Windows. В частности, она позволяет задать задержку перед запуском и количество потоков шифрования. Некоторые возможности этой сборки зловреда совпадают с функциями Windows-версии: настройка процента шифруемых файлов, выбор целевого каталога и указание пути к лог-файлу.

Шифрование файлов

Схема шифрования файлов идентична версии для Windows. Троянец использует алгоритм XChaCha20-Poly1305 для шифрования содержимого файлов и метаданных, а также Curve25519-XSalsa20-Poly1305 для шифрования ключа.

```
GENIE LOCKER v1 ESXI x64

Max workers: 4
Head:        96 MB
Pct:         50%
Extension:    .4843c5aca9f93182

Found 21 file(s). Parallel workers: 2
```

Сводка по процессу шифрования файлов

Жертвы

По данным телеметрии KSN, подавляющее большинство детектирований GenieLocker приходится на конечные устройства, расположенные в Российской Федерации. В ходе кампании в марте 2026 года наибольшее количество атак пришлось на производственные предприятия. С небольшим отрывом за ними следуют строительные компании, а затем организации из сферы финансовых услуг, розничной и технологической отраслей.

Заключение

Кампания группы Toy Ghoul, направленная против российских предприятий, продолжает набирать обороты. Переход на собственный шифровальщик GenieLocker стал значительным обновлением арсенала вымогательского ПО группы. Разработав специализированный инструмент с нативными версиями для Windows, Linux и ESXi, злоумышленники снизили зависимость от готовых семейств шифровальщиков и унифицировали криптографическую основу для атак на эти целевые платформы.

Продукты «Лаборатории Касперского» детектируют это вредоносное ПО как Trojan-Ransom.Win64.Agent.genie, HEUR: TrojanRansom.Win64.Generic и Trojan-Ransom.Linux.Agent.genie.

Индикаторы компрометации

Дополнительная информация об этой угрозе доступна клиентам [сервиса Kaspersky Threat Intelligence Reporting](#). Для получения подробной информации свяжитесь с нами по адресу intelreports@kaspersky.com.

GenieLocker для Windows

[A50EAAF514F4F84E61CA2455A8789753](#) kftd.exe, genie_encrypt.exe
[F08F476F26B01D142CA73923DE65FC0C](#)
[FD46A80C2F45577263328984EDF7F4DC](#)
[DE3CFBB50F66079BFEE20A6F64E59433](#)
[780C8F4C6F077DA4DA96582987920362](#)
[D87D0B01D95ACC936B7DC47B8F41937A](#) run.exe, genie_encrypt.exe
[34A7F28E0BB69B0D49BACC88BDF20AC1](#) run.exe, run2.exe, genie.exe
[5D62C1349B8981C396C9A23F4F8F053C](#) genie_encrypt.exe
[A8842616C9057D5CF6E1FE1FA8C3C160](#)
[34B8828635F88078735799A3C1AC8E28](#)
[D3E06EB34D8EEE7EF92CAC3AD0A20FF5](#)
[C68B6862725777651085650DB34947FC](#) consultant.exe
[9CD514FF2809CE0B993E3B8649E82A94](#)
[824CA1E906CC073EE5B0F3519DF69A8F](#)
[25480DAD40152EF3D0C6D38EECC9BD9B](#)
[7DAD78584795AA5C160520CC6ACCF260](#)
[18F61C6D686CFFD131C9FD3F3437064B](#) tempo.exe, kernel.exe
[9969A8221312DBA70DD5CBDDF83A146C](#)
[F7B9E36E94163A9A303160945F99267A](#)
[B893EAFED0659F70D4AC250F09073723](#)

[D661CF666B9ACBAB7CFEAE1127A261A9](#) genie.exe

[3A4479B51890373BFC4A011EF41FE376](#)

[58C0DDA52B8F069660166D61FD74F911](#)

GenieLocker для Linux и ESXi

[9201E35E2993612612919A3C71302CAB](#) vzdump

C2

[89\[.\]125.66.101](#)

- [Технологии вредоносных программ](#)
- [Описание вредоносных программ](#)
- [Microsoft Windows](#)
- [Linux](#)
- [Вредоносные программы](#)
- [Троянец-шифровальщик](#)
- [Вымогатели](#)
- [Шифрование](#)
- [ESXi](#)