Статья Вымогатель REvil-как-Сервис: Анализ партнерской операции вымогателей

Резюме

REvil, также известный как Sodinokibi, Sodin - это семейство программ-вымогателей, работающее как программа-вымогатель как услуга (RaaS). Развертывание REvil впервые было замечено в апреле 2019 года, когда злоумышленники использовали уязвимость в серверах Oracle WebLogic, именуемую как CVE-2019-2725.

REvil легко конфигурируется и позволяет операторам настраивать его поведение на зараженном хосте. Некоторые из его функций включают:

> - Эксплуатирует уязвимость повышения привилегий ядра для получения привилегий SYSTEM с помощью CVE-2018-8453.
> - Делает белые списки файлов, папок и расширений защищая их от шифрования.
> - Убивает определенные процессы и службы до шифрования.
> - Шифрует файлы в локальном и сетевом хранилище.
> - Настраивает имя и текст записки с требованием выкупа, а также содержимое фонового изображения.
> - Передает зашифрованную информацию о зараженном хосте на удаленные контроллеры.
> - REvil использует защищенный протокол передачи гипертекста (HTTPS) для связи со своими контроллерами.

Обзор возможностей

- Программы-вымогатели

Обзор поведения

> - Использует механизм закрепления в системе.
> - Шифрует дополнительные ресурсы.
> - Поддерживает повышение привилегий.

Разведка противника
Разработчики

Программа-вымогатель REvil впервые была разрекламирована на русскоязычном форуме по киберпреступности в июне 2019 года. Основное лицо, связанное с рекламой и продвижением вымогателя REvil, носит имя Unknown aka UNKN. RaaS работает как партнерская служба, где партнеры распространяют вредоносное ПО, захватывая жертв, а операторы REvil обслуживают вредоносное ПО и платежную инфраструктуру. Партнеры получают от 60% до 70% суммы выкупа.

Из-за сходства исходного кода и поведения REvil и GandCrab было высказано предположение, что существует связь, связывающая разработчиков двух семейств программ-вымогателей. Помимо сходства в коде, дополнительным свидетельством, связывающим GandCrab и REvil, является то, что GandGrab официально "ушел на пенсию" незадолго до того, как REvil появился на свет. REvil активно поддерживается и постоянно развивается, как и GandCrab. Самым последним вымогателем REvil на момент написания этого отчета была версия 2.1.

Главное лицо представляющее шифровальщик — Unknown, подтвердил публичные сообщения, связывающие REvil с GandCrab, следующим заявлением:

"**Раньше мы были участниками партнерской программы (GandCrab). Мы купили исходный код и открыли собственный бизнес. Мы разработали специальные функции для наших целей**".

Несмотря на правдоподобное алиби, данное Unknown, данные свидетельствуют о том, что REvil является продолжением операции GandCrab RaaS с новым программным обеспечением, но управляется теми же людьми.

Люди, стоящие за REvil, включают свой главный открытый ключ во все двоичные файлы Revil. Таким образом, они могут расшифровать файлы независимо от партнеров участвующих в кампании.

Операторы

REvil - это RaaS в том смысле, что одна группа работает и управляет разработкой вымогателей, в то время как доступ продается аффилированным лицам. Поле в файле конфигурации с именем pid используется для идентификации аффилированного лица, которому принадлежит образец. Мы подтвердили, что поле в конфигурации вредоносного ПО с именем sub используется для идентификации партнерских кампаний, а не самих аффилированных лиц, как часто сообщается. Кроме того, открытый ключ злоумышленника может идентифицировать одного и того же оператора при использовании нескольких образцов.

Два известных пользователя вышеупомянутого русскоязычного форума поручились за службу вымогателей Unknown:

- kerberos - Модератор вышеупомянутого форума и давний киберпреступник.

- lalartu - киберпреступник, известный своим участием в партнерских программах вымогателей GandCrab и Revil.

Личная информация человека lalartu, возможно, была разглашена в злонамеренных целях или "закреплена" исследователем информационной безопасности, известным как UnderTheBreach (https://medium.com/@underthebreach/tracking-down-revils-lalartu-by-utilizing-multiple -osint-methods-2bf3a6c65a80).

Мы не смогли проверить выводы, сделанные в этом сообщении.

Инфраструктура

Конфигурации вымогателя REvil содержат более 1000 контроллеров. Интересно, что все проверенные нами действующие домены были веб-сайтами WordPress, поэтому вполне вероятно, что они могут быть взломаны операторами или приобретены у других киберпреступников.

Конфигурация также содержит доменные имена, которые не были зарегистрированы на момент составления этого отчета, например:

$ whois andersongilmour.co.uk

No match for “andersongilmour.co.uk.”

Это доменное имя не было зарегистрировано на момент составления настоящего отчета.

Скорее всего, большинство доменов, присутствующих в конфигурации, являются ловушками, и только несколько реальных контроллеров REvil разбросаны внутри списка.

Обнаружение

В этом разделе описаны методы, используемые для обнаружения сэмпла Revil.

Статическое обнаружение

Распакованные образцы REvil могут быть обнаружены статически путем поиска шаблонов в коде и в криптографических функциях, используемых программой-вымогателем.

Динамическое обнаружение

Динамическое обнаружение сэмплов REvil возможно путем запуска подписей Yara на резидентном образе памяти. Кроме того, программа-вымогатель создает следующие интересные артефакты:

> - Файл .txt с запиской о выкупе в каждом каталоге, зашифрованном программой-вымогателем.
> - Изображение .bmp во временном каталоге, установленном в качестве фона рабочего стола после шифрования.
> - Ключ реестра SOFTWARE, которое может находиться в HKEY_LOCAL_MACHINE или HKEY_CURRENT_USER.
> - Буквенно-цифровое расширение файла длиной от 5 до 10 символов, добавленное к исходному расширению зашифрованного файла.

Сигнатуры Yara

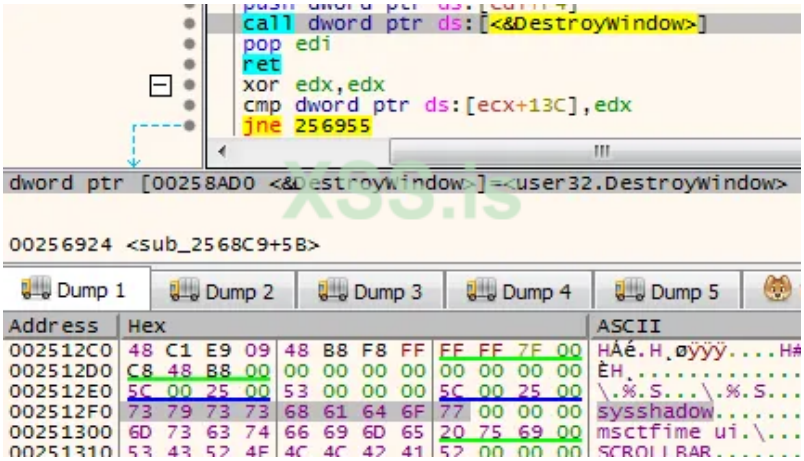
Мы решили не выпускать сигнатуры Yara, чтобы не повлиять на работу других исследователей. Тем не менее, мы передадим их защитникам сети и специалистам в области информационной безопасности по запросу. Напишите нам по адресу revil-yara-req@intel471.com со своего

корпоративного адреса электронной почты (только в целях проверки), и мы отправим вам подписи через нашу Yara.

Эксплойты

Программа-вымогатель REvil эксплуатирует уязвимость повышения привилегий ядра в файле win32k.sys, отслеживаемую как CVE-2018-8453, для получения системных привилегий на зараженном узле. Если выполнить этот эксплойт, он выделит исполняемую память, расшифрует код эксплойта во вновь выделенной области и вызовет его.

На снимке экрана ниже показано имя окна sysshadow и интерфейс прикладного программирования (API) DestroyWindow внутри встроенного эксплойта, что доказывает, что это уязвимость CVE-2018-8453.



Open Source Intelligence (OSINT)

Для получения дополнительной информации о REvil мы предлагаем следующие общедоступные ресурсы:

- REvil/Sodinokibi Ransomware (https://www.secureworks.com/research/revil-sodinokibi-ransomware).

- McAfee ATR Analyzes Sodinokibi aka REvil Ransomware-as-a-Service – What The Code Tells Us (https://www.mcafee.com/blogs/other-...nsomware-as-a-service-what-the-code-tells-us/).

- Tracking REvil (https://www.kpn.com/security-blogs/Tracking-REvil.htm).

- Defeating Sodinokibi/REvil String-Obfuscation in Ghidra (https://blag.nullteilerfrei.de/2020/02/02/defeating-sodinokibi-revil-string-obfuscation-in-ghidra/).

Статический анализ

Программа-вымогатель REvil включает методы, которые усложняют задачу статического анализа для аналитика. Большинство строк, используемых во время выполнения, расшифровываются во время выполнения только при необходимости. Кроме того, импорт

динамически разрешается из 32-битных хэшей и помещается в глобальные переменные на раннем этапе выполнения.

Жестко запрограммированные строки

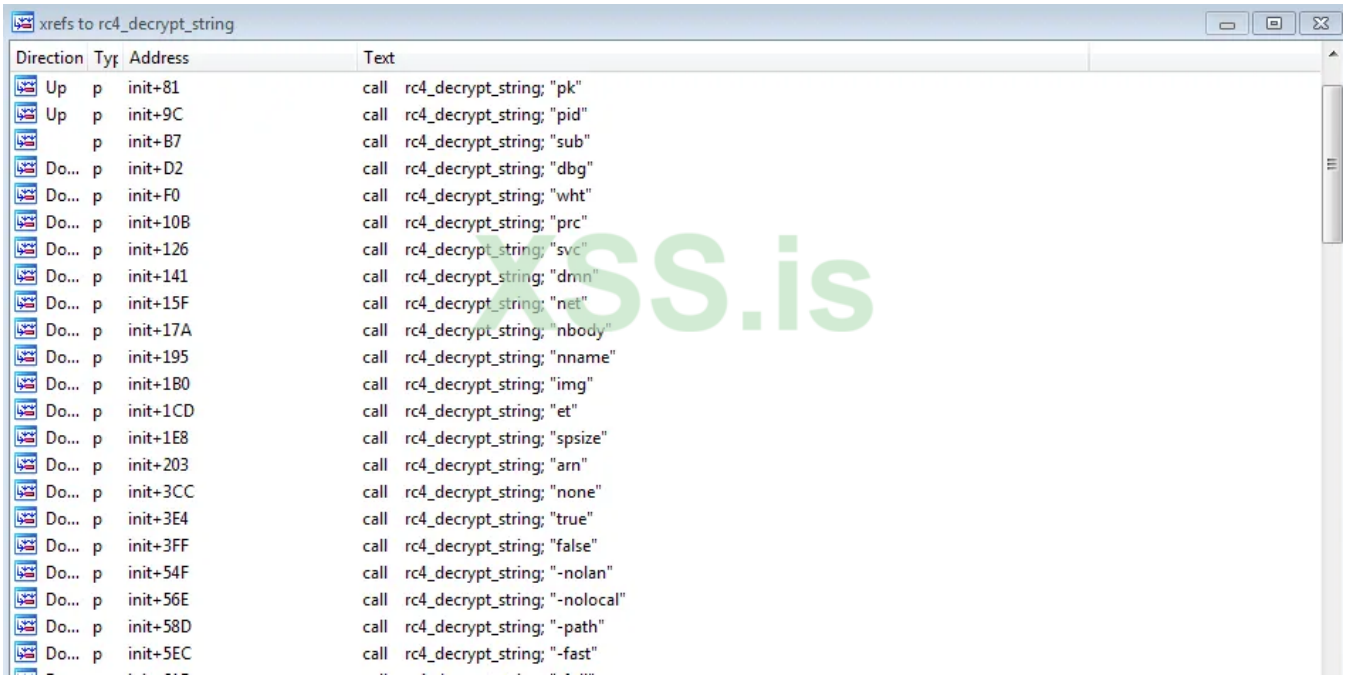
Очень мало строк жестко закодированы в образцах вымогателя REvil, и наиболее интересными являются:

- L"ServicesActive": эта строка передается в API OpenSCManagerW для получения активных служб.

- "expand 32-byte kexpand 16-byte": константы, используемые алгоритмом симметричного шифрования Salsa20.

Шифрование строки

Программа-вымогатель REvil расшифровывает большую часть строк, используемых во время выполнения, во время запуска.



Как показано на скриншоте выше, эти строки зашифрованы Rivest Cipher 4 (RC4) и расшифровываются с помощью функции, которую мы переименовали в rc4_decrypt_string. Ниже представлен его прототип и значение каждого параметра:

void rc4_decrypt_string(BYTE *rc4_array, int rc4_key_offset, int rc4_key_length, int buffer_length, BYTE *out_buffer)

- rc4_array: указатель на непрерывный массив в разделе .data, содержащий ключи RC4 и зашифрованные строки. За каждым ключом RC4 следует строка, которую он расшифровывает.

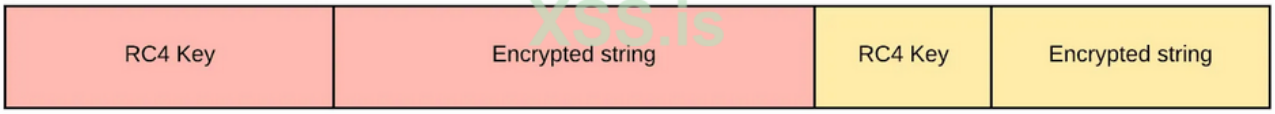
- rc4_key_offset: смещение ключа RC4 в массиве.

- rc4_key_length: длина ключа RC4.

- buffer_length: длина зашифрованного буфера RC4.

- out_buffer: предварительно выделенная память или пространство стека, куда копируется расшифрованная строка.

На изображении ниже показано расположение двух смежных элементов массива:



Расшифрованные строки не заканчиваются символами NULL, поэтому код должен явно завершать эти строки.

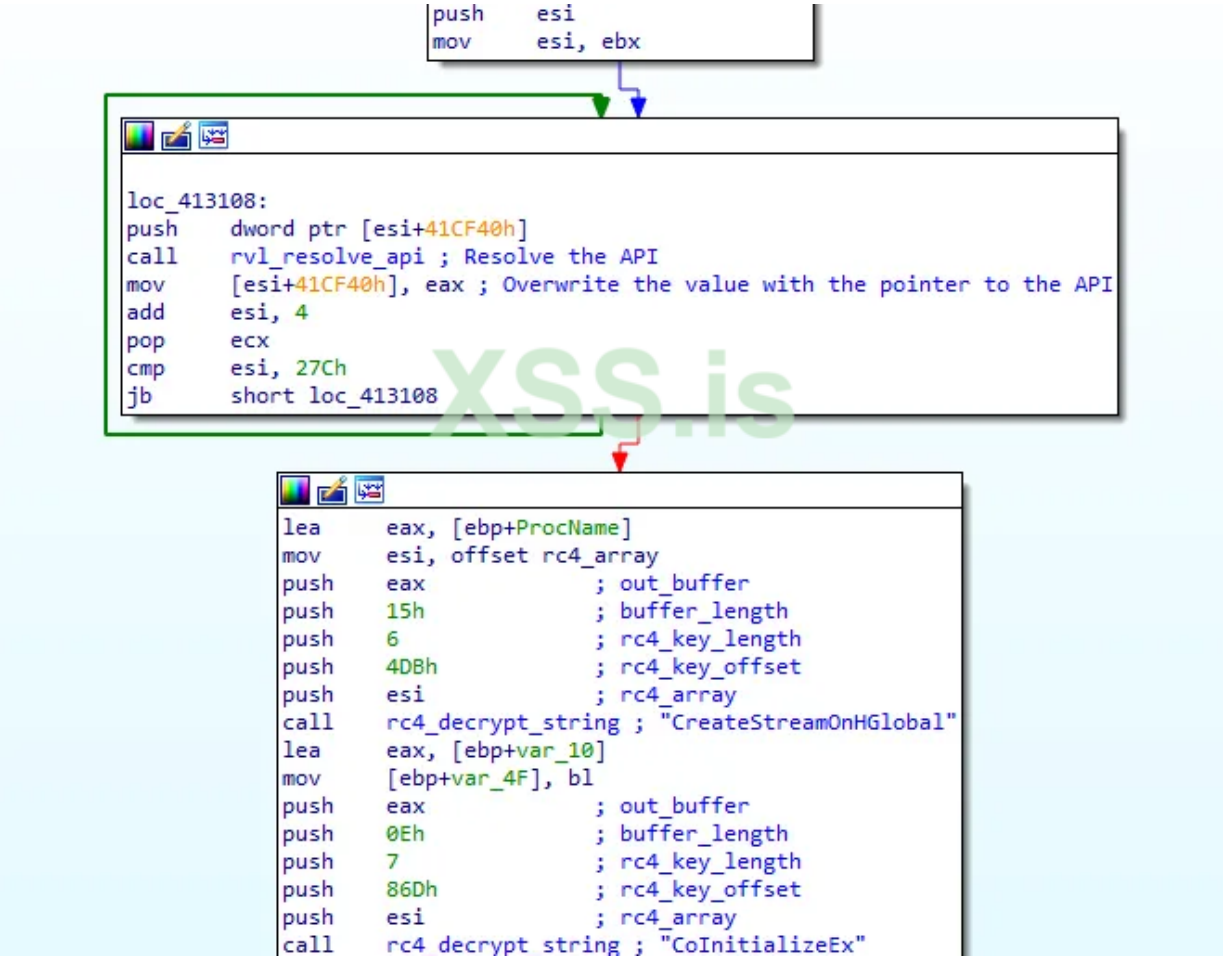
Поведение вредоносного ПО

Для этого отчета было проанализировано поведение следующих образцов:

Сэмпл	SHA256
REvil упакованный	6953d86d09cb8ed34856b56f71421471718ea923cd12c1e72224356756db2ef1
REvil не упакованный	372c8276ab7cad70ccf296722462d7b8727e8563c0bfe4344184e1bc3afc27fc
REvil не упаковынный	ec0c653d5e10fec936dae340bf97c88f153cc0cdf7079632a38a19c876f3c4fe

Выполнение процесса

Во время фазы инициализации REvil начинает с динамического разрешения импорта, необходимого для правильной работы. Это выполняется в цикле, который считывает жестко закодированные 32-битные значения из массива в разделе .data, затем каждое значение декодируется и преобразуется в правильный API с помощью ответственной функции rvl_resolve_api. Затем 32-битное значение перезаписывается указателем на API.

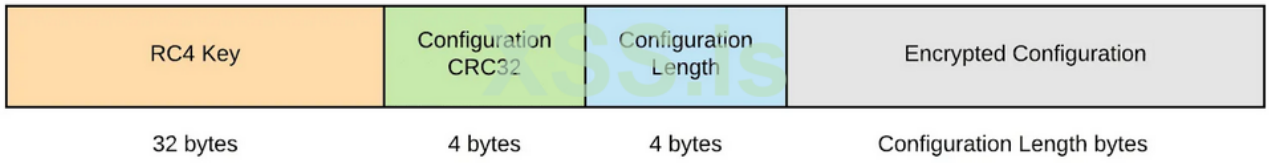


Как показано в дизассемблерном коде выше, дополнительные API-интерфейсы разрешаются по их именам с помощью API GetProcAddress.

Когда все готово, REvil создает глобальный объект взаимного исключения (мьютекс) с жестко заданным именем, например Global\1DE3C565-E22C-8190-7A66-494816E6C5F5. Это используется, чтобы гарантировать, что запущен только один экземпляр образца программы-вымогателя.

Программа-вымогатель REvil всегда пытается работать с повышенными привилегиями и делает это двумя способами. Один метод основан на использовании уязвимости CVE-2018-8453 для получения привилегий SYSTEM на хосте. Однако, чтобы определить, следует ли запускать эксплойт, REvil проверяет его конфигурацию. Другой прием всегда выполняется, если процесс не повышен. Он полагается на вызов ShellExecuteW, чтобы предложить пользователю запустить образец от имени администратора. Это выполняется в бесконечном цикле, пока пользователь не согласится запустить процесс с повышенными правами. Начиная с версии 2.1, код эксплойта повышения привилегий был удален.

Конфигурация REvil используется в объектной нотации JavaScript (JSON) и изначально зашифрована RC4. Он хранится следующим образом в начале раздела переносимого исполняемого файла (PE) с именем .yhwfq9:



Важно отметить, что значение циклического контроля избыточности (CRC32) вычисляется и проверяется для зашифрованной конфигурации.

После расшифровки конфигурации в динамически выделяемой памяти REvil анализирует ее, используя библиотеку C json-parser с открытым исходным кодом (https://github.com/udp/json-parser). Пример конфигурации REvil смотри Приложения ниже.

Первое проверяемое поле JSON - это exp, которое может иметь значение true или false и указывает, следует ли использовать уязвимость CVE-2018-8453. Если эксплойт не выполняется или завершился ошибкой, REvil прибегает ко второму способу, ранее описанному в этом разделе.

Когда REvil выполняется с более высокими привилегиями, он начинает свою основную фазу инициализации, на которой он считывает необходимые поля JSON в раздел .data и инициализирует значения реестра внутри нового подраздела с именем SOFTWARE.

При выполнении операций с реестром в целом REvil сначала пытается использовать куст HKEY_LOCAL_MACHINE и переключается на использование HKEY_CURRENT_USER, только если это не удается. Поля конфигурации, разделы реестра и их значения описаны в разделе Configuration ниже.

Другой важной задачей на этом этапе является заполнение шаблона примечания с требованием выкупа, который присутствует в base64 внутри конфигурационного поля nbody.

--- === Добро пожаловать. В очередной раз. === ---

[+] Что случилось? [+]

Ваши файлы зашифрованы и в настоящее время недоступны. Вы можете это проверить: все файлы на вашем компьютере имеют расширение {EXT}. Кстати, все можно восстановить (восстановить), но нужно следовать нашим инструкциям. В противном случае вы не сможете вернуть свои данные (НИКОГДА).

[+] Какие гарантии? [+]

Это просто бизнес.

Мы абсолютно не заботимся о вас и ваших сделках, кроме получения выгоды.

Если мы не будем выполнять свою работу и обязательства - никто с нами не будет сотрудничать. Это не в наших интересах. Чтобы проверить возможность возврата файлов, Вам следует зайти на наш сайт. Там можно бесплатно расшифровать один файл. Это наша гарантия. Если вы не будете сотрудничать с нашим сервисом - для нас это не имеет значения. Но вы потеряете свое время и данные, потому что только у нас есть закрытый ключ. На практике время гораздо дороже денег.

[+] Как получить доступ на сайт? [+]

У вас есть два пути:

1) [Рекомендуется] Используйте браузер TOR!
а) Загрузите и установите браузер TOR с этого сайта: https://torproject.org/
б) Откройте наш сайт:
hxxp://aplebzu47wgazapdqks6vrcv6zcnjppkbxbr6wketf56nf6aq2nmyoyd. onion/{UID}

2) Если TOR заблокирован в вашей стране, попробуйте использовать VPN! Но вы можете использовать наш вторичный сайт. Для этого:
а) Откройте любой браузер (Chrome, Firefox, Opera, IE, Edge)
б) Откройте наш дополнительный веб-сайт: hxxp://decryptor.cc/{UID}

Предупреждение: второй сайт может быть заблокирован, поэтому первый вариант намного лучше и доступнее. Когда вы открываете наш сайт, введите следующие данные в форму ввода:

{KEY}

Extension name:

{EXT}

!!! ОПАСНОСТЬ !!!

НЕ пытайтесь изменять файлы самостоятельно, НЕ используйте какое-либо стороннее программное обеспечение для восстановления ваших данных или антивирусные решения - это может повлечь за собой повреждение закрытого ключа и, как следствие, потерю всех данных.

!!! !!! !!!
ЕЩЕ ОДИН РАЗ. В ваших интересах вернуть свои файлы. Со своей стороны мы (лучшие специалисты) делаем все для восстановления, но нам не должны мешать.
!!! !!! !!!

REvil заполняет этот шаблон расширением файла (EXT), идентификатором пользователя (UID) и ключом (KEY):

- Расширение генерируется случайным образом из буквенно-цифровых символов и имеет длину от 5 до 10 символов.

- Идентификатор пользователя - это идентификатор оборудования, созданный на основе серийного номера основного тома и информации о процессоре системы. Этот идентификатор рассчитывается следующим образом:

> - Запрашивается 32-битный серийный номер основного тома.
> - Создаётся контрольная сумму CRC32 для серийного тома с начальным значением, равным 1337.
> - Запрашивается информация о центральном процессоре (ЦП), например, "Intel (R) Core (TM) i7-6700HQ CPU @ 2,60 ГГц".
> - Генерируется контрольная сумма CRC32 информационной строки ЦП с хешем CRC32 последовательного тома в качестве начального значения.
> - Объединяется хэш CRC32 информации о ЦП с серийным номером тома в 16-байтовую шестнадцатеричную строку, то есть 8100F233BC097E90.

Ключ, который жертва передает оператору через веб-сайт, представляет собой строку JSON:

{

"ver":"%d", # Версия REvil: жестко запрограммирована на 0x200 (v2.00).

"pid":"%s", # Поле "pid" в конфигурации.

"sub":"%s", # Поле "sub" в конфигурации.

"pk":"%s", # Поле открытого ключа "pk" в конфигурации.

"uid":"%s", # Идентификатор оборудования.

"sk":"%s", # Сгенерированный жертвой секретный ключ, зашифрованный открытым ключом "pk". Этот ключ необходим для расшифровки.

"unm":"%s", # Имя пользователя учетной записи Windows.

"net":"%s", # Имя компьютера.

"grp":"%s", # Доменное имя.

"lng":"%s", # Язык т.е. "Fr-FR".

"bro":%s, # Невосприимчив к инфекциям, язык из белого списка и раскладка клавиатуры.

"os":"%s", # Название продукта Windows.

"bit":%d, # Архитектура процессора: x86 или 64.

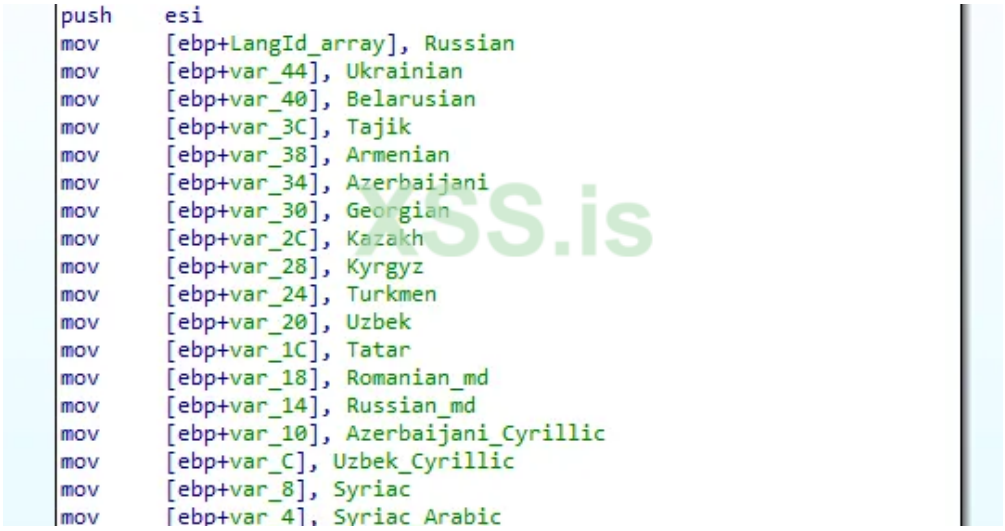
"dsk":"%s", # Структура в кодировке base64 с информацией о смонтированных томах.

"ext":"%s" # Расширение зашифрованных файлов, то есть ".g19b9wy".

}
Click to expand...

Он шифруется жестко закодированным открытым ключом в двоичном коде перед сохранением в реестре и шаблоне. Это также передается контроллерам позже при исполнении. Смотри раздел Шифрование ниже для получения информации о том, как REvil шифрует данные.

Затем REvil проверяет конфигурационное поле dbg, чтобы увидеть, работает ли оно в режиме отладки. Если это не так, выполняется проверка геолокации на основе языка системы и раскладки клавиатуры, чтобы программа-вымогатель не пыталась зашифровать файлы в системах из белого списка. Следующие идентификаторы системного языка занесены в белый список для анализируемого образца:



Раскладки клавиатуры из белого списка включают

> Румынский
> Русский
> Украинец
> Белорусский
> Эстонский
> Латышский
> Литовский язык
> Таджикский
> Персидский
> Армянский
> Азербайджанский
> Грузинский
> Казахский
> Кыргызский
> Туркменский
> Узбекский
> Татарский
>
> Click to expand...

Для успешной проверки и выхода REvil должны присутствовать как язык системы из белого списка, так и раскладка клавиатуры из белого списка. В противном случае программа-вымогатель продолжит работу в обычном режиме.

На этом этапе REvil выполняет следующие действия перед запуском шифрования. Во-первых, он попытается остановить и удалить службы, если имена совпадают с одним из регулярных выражений в списке конфигурации svc JSON, например:

> **"svc":**[
> "memtas",
> "crm",
> "quickbooks",
> "svc$",
> "veeam",
> "oracle",
> "mepocs",
> "exchange",
> "pos",
> "vss",
> "sql",
> "backup",
> "qb",
> "sophos",
> "sage"
>]
>
> Click to expand...

Затем он завершит все процессы с именами, соответствующими элементам массива prc JSON, например:

> **"prc":**[
> "w3wp",
> "thunderbird",
> "mydesktopqos",
> "powerpnt",
> "outlook",
> "srv",
> "infopath",
> "msaccess",
> "ocautoupds"
>]

Наконец, REvil удаляет теневые копии тома. Способ выполнения этого зависит от версии Windows:

Windows 5.1 и ранние:

cmd.exe /c vssadmin.exe Delete Shadows /All /Quiet & bcdedit /set {default} recoveryenabled No & bcdedit /set {default} bootstatuspolicy ignoreallfailures

Версии Windows 5.2 и новее в PowerShell:

Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}

На этом этапе REvil переходит в фазу шифрования файла. REvil можно запускать с аргументами командной строки, которые повлияют на то, как выполняется шифрование. В следующей таблице описаны эти аргументы:

Аргумент Описание

> -full Шифрует все данные в целевых файлах.
> -fast Шифрует только первый МБ каждого файла. Взаимоисключающие с аргументом -full.
> -path Рекурсивно шифрует файлы внутри одного каталога, указанного после аргумента. При указании этого аргумента фоновое изображение рабочего стола остается неизменным.
> -nolan Не шифрует файлы в общем сетевом хранилище.
> -nolocal Не шифрует файлы в локальном хранилище.

Два значения, которые может принимать поле конфигурации et, эквивалентны аргументам командной строки -full и -fast. Кроме того, наличие одного из этих аргументов командной строки переопределяет значение этого поля. Ниже приведены значения, которые может принимать это поле, с описанием каждого значения:

> Значения типа шифрования Описание
> 0 Эквивалентно аргументу командной строки -full
> 1 Эквивалентно аргументу командной строки -fast
> 2 Шифрует 1 МБ, затем повторно пропускает количество МБ, указанное в поле spsize, пока не будет достигнут конец файла.

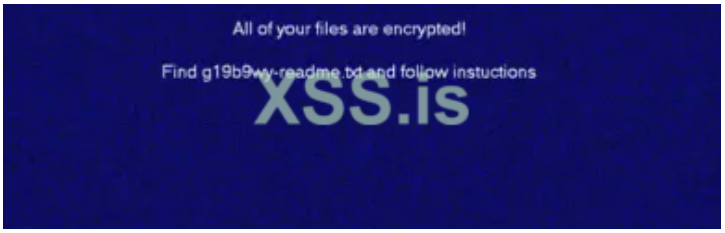
Перед шифрованием файла или каталога REvil определяет, соответствует ли его имя какой-либо записи в записях конфигурации белого списка.Папки, файлы и расширения из белого списка - это списки, хранящиеся в поле конфигурации wht JSON как fld, fls и ext.

Программа-вымогатель REvil также избегает шифрования файла более одного раза, определяя, был ли он ранее зашифрован. Это достигается путем проверки того, существуют ли метаданные, хранящиеся в конце зашифрованных файлов.

Для шифрования файлов REvil использует многопоточность и порты завершения ввода-вывода (I / O), что позволяет ему выполнять асинхронный ввод-вывод и обрабатывать несколько файлов одновременно.

Каждый файл зашифрован другим ключом, полученным из одного открытого ключа, связанного с жертвой.

После завершения шифрования REvil генерирует растровое изображение и устанавливает его в качестве фона рабочего стола.



После этого программа-вымогатель REvil считывает сеть логического поля конфигурации, чтобы определить, следует ли ей связываться со своими контроллерами. Смотри раздел Связь ниже для получения дополнительных сведений.

Наконец, программа-вымогатель REvil помечает свой двоичный код для удаления при следующей перезагрузке и прекращает выполнение.

Конфигурация

В таблице ниже описано каждое поле конфигурации REvil JSON:

Аргумент Описание

pk - Открытый ключ злоумышленника, закодированный в base64.
pid - Партнерский идентификатор. В версии 2.0 и ранее это было целое число. Начиная с версии 2.1 он стал хешем Bcrypt.
sub - Целочисленное значение. Идентификатор кампании.
dbg - Логическое значение, определяющее, должен ли REvil работать в режиме отладки.
et - Целочисленное значение, определяющее тип шифрования. - 0: быстрое шифрование. - 1: полное шифрование. - 2: зашифровать 1 МБ, затем пропустить количество МБ, указанное в поле spsize.
spsize - Задает количество пропускаемых МБ, если тип шифрования равен 2.
wipe - Логическое значение, указывающее, должен ли REvil стирать папки, указанные в элементе wfld. В проанализированных образцах мы не видели реализации этой опции.
wfld - Папки, которые будет очищать REvil.
wht - Содержит три списка элементов, которые REvil не будет шифровать: - fld: папки из белого списка. - fls: файлы из белого списка. - ext: расширения из белого списка.
prc - Список регулярных выражений, которые REvil сопоставляет с процессами для их завершения.
net - Если это логическое значение установлено в true, REvil взаимодействует со своими контроллерами.
dmn - Строка контроллеров, разделенных точкой с запятой.
svc - Список регулярных выражений для сопоставления с запущенными службами для их остановки и удаления.
nbody - Шаблон записки с требованием выкупа, закодированный в base64.
nname - Имя файла с запиской о выкупе, которая будет помещена в зашифрованные каталоги.
exp - Указывает, следует ли REvil использовать уязвимость CVE-2018-8453 для повышения привилегий.
img - Текст, который будет записан в фоновом изображении, закодированный в base64.
arn - Если установлено значение true, REvil сохраняется в системе.

Click to expand...

Значения реестра, которые создает REvil, описаны ниже:

Имя значения Описание

1TfXk Секретный ключ жертвы зашифрован открытым ключом злоумышленника, присутствующим в конфигурации.
2YEdLY Секретный ключ жертвы зашифрован главным открытым ключом, жестко закодированным в двоичном коде.
aah Открытый ключ злоумышленника.
fdle Открытый ключ жертвы.
AaZW1s3 Расширение, добавляемое к файлам после шифрования.
QaUXNv2P Ключ жертвы зашифрован вторым жестко закодированным открытым ключом в двоичном коде.

Шифрование

Программа-вымогатель REvil реализует схемы шифрования с использованием эллиптической кривой Curve25519, Salsa20, SHA-3, CRC32 и Advanced Encryption Standard (AES).

Мы определили конкретные реализации с открытым исходным кодом, используемые Revil:

> - Curve25519: https://github.com/vstakhov/opt-cryptobox/tree/master/curve25519 .
> - Salsa20: https://cr.yp.to/snuffle/salsa20/merged/salsa20.c .

В этом подразделе подробно описывается, как эти алгоритмы используются для шифрования данных и файлов.

Ключи шифрования

Программа-вымогатель REvil использует Curve25519 для генерации пар открытых и секретных ключей и для создания общих ключей для шифрования.
Ниже приводится определение ключей Curve25519, упомянутых в этом подразделе:

> Название пары ключей Описание
>
> **crypt** (crypt_public/crypt_secret) - Пара основных ключей жертвы, используемая для шифрования файлов.
>
> **file** (file_public/file_secret) - Случайная пара ключей, созданная для каждого зашифрованного файла.
>
> **attacker** (attacker_public/attacker_secret) - Пара ключей атакующего. Attacker_public присутствует в поле конфигурации pk.
>
> **master** (master_public/master_secret) Пара мастер-ключей. Master_public жестко закодирован внутри двоичного кода и одинаков для всех образцов REvil.
>
> **user_config** (user_config_public/user_config_secret) - Пара ключей конфигурации пользователя. User_config_public жестко закодирован в двоичном коде и используется для шифрования пользовательского ключа, который мы находим зашифрованным в записке о выкупе.
> Click to expand...

Шифрование данных

Программа-вымогатель REvil шифрует все важные данные, хранящиеся в реестре. Например, пользовательский ключ, присутствующий в записке о выкупе и в реестре, представляет собой словарь JSON, зашифрованный с использованием метода, который мы описываем в следующих параграфах.

Чтобы зашифровать буфер, REvil вызывает следующую функцию:

BYTE* rvl_encrypt_data(BYTE *hispublic, BYTE *buffer, int buffer_length, BYTE *out_buffer_length)

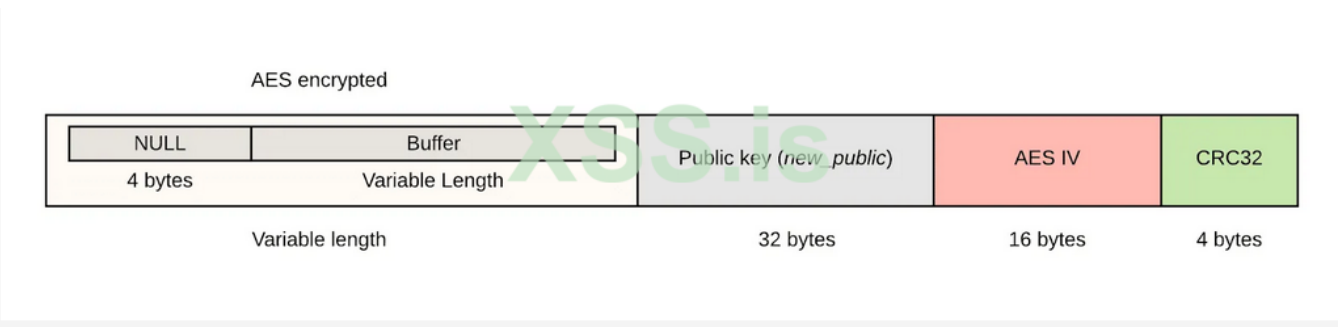
Эта функция требует 32-байтового открытого ключа Curve25519, буфера для шифрования и его длины. Конечным результатом является указатель на буфер в возвращаемом значении и его длина в выходном параметре out_buffer_length.

Внутренне эта функция выполняет следующие действия:

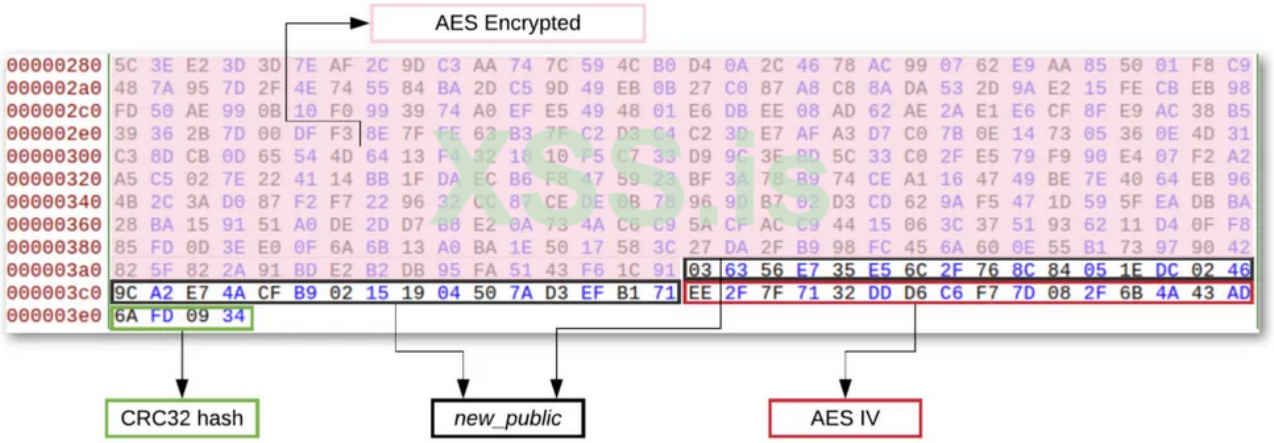
> - Выделяет buffer_length байтов плюс дополнительные 56 байтов для хранения окончательных данных.
> - Обнуляет первые 4 байта выделенного буфера.
> - Копирует буфер для шифрования после обнуленного двойного слова.
> - Создаёт новую пару ключей Curve25519, которую мы называем new_public и new_secret.
> - Вычисляет общий ключ Curve25519 между его открытым ключом, указанным в аргументах, и new_secret. Получатель сообщения может использовать секретный ключ и new_public для получения одного и того же общего ключа.
> - Хеширует общий ключ с помощью алгоритма SHA-3.
> - Очищает общий ключ из памяти.
> - Очищает new_secret из памяти.
> - Генерирует случайный 128-битный вектор инициализации AES.
> - Зашифровывает буфер с помощью AES с хешем SHA-3 в качестве ключа. Буфер для шифрования включает в себя добавленное 32-битное значение NULL.
> - Очищает хэш SHA-3 из памяти.
> - Вычисляет CRC32 зашифрованного буфера.
> - Добавляет в зашифрованный буфер следующие элементы в указанном порядке:
>
> Click to expand...

Сгенерированный ключ new_public (32 байта).
Вектор инициализации (16 байт).
CRC32 зашифрованного буфера (4 байта).

Конечный результат выглядит так:



Например, после декодирования пользовательского ключа, присутствующего в записке с требованием выкупа, с использованием base64, мы видим, что он соблюдает тот же формат:



Чтобы злоумышленники могли расшифровать эти данные и получить словарь JSON и, как результат, зашифрованный ключ crypt_secret жертвы, они должны сделать следующее:

> - Проверить хэш CRC32 зашифрованных данных.
> - Вычислить общий ключ Curve25519, используя новый_публичный ключ в данных и секрет ключ злоумышленника user_config_secret.
> - Хешировать общий ключ с помощью SHA-3.
> - Получить вектор инициализации AES и расшифровать буфер с помощью AES.
> - Удалить двойное слово NULL в начале буфера.

Важно отметить, что ключ crypt_secret жертвы зашифрован таким же образом с использованием ключа атакующего_публичного (значение реестра «1TfXk») и ключа master_public (значение реестра «2YEdLY»).

Шифрование файлов

Шифрование программ-вымогателей REvil работает путем шифрования файлов с шагом 1 МБ, если размер файла не меньше, оставшиеся байты для шифрования меньше 1 МБ или если указанный тип шифрования не является полным шифрованием. Содержимое файла читается, шифруется и записывается обратно в файл, перезаписывая исходное содержимое. После завершения шифрования метаданные записываются в конец файла, и файл переименовывается, чтобы включить сгенерированное расширение файла (значение реестра AaZW1s3).

Для каждого данного файла вымогатель REvil запускает шифрование, инициализируя структуру, используемую на протяжении всего процесса.
Его первая часть включает структуру Windows OVERLAPPED, используемую для асинхронного ввода-вывода, за которой следуют настраиваемые поля, используемые REvil. Самая интересная часть этой структуры была реконструирована следующим образом:

C:

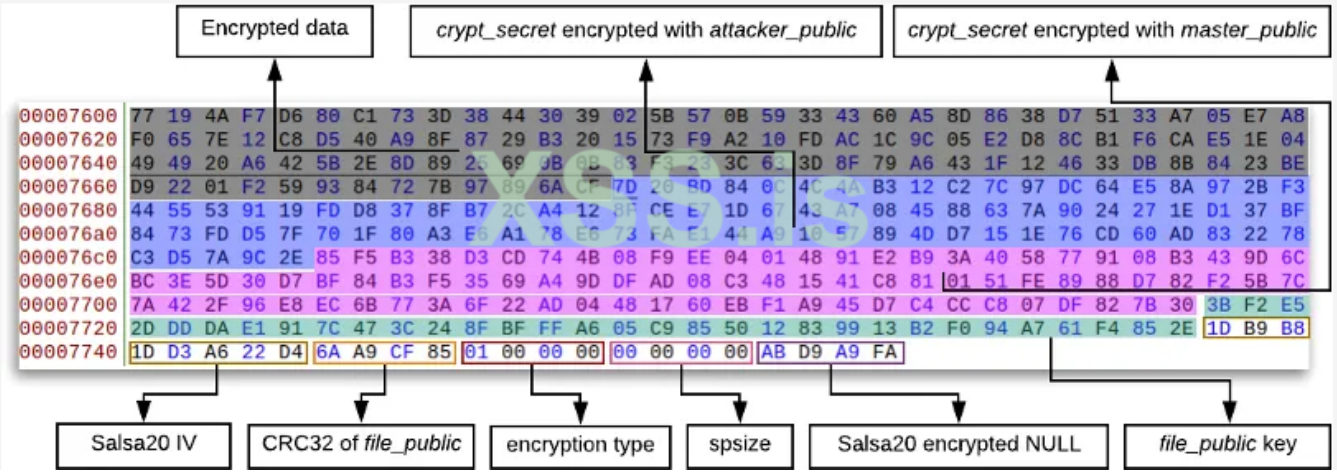
```
typedef struct _rvl_filecrypt_struct
{
OVERLAPPED Overlapped;
/*..SNIP..*/
struct _rvl_metadata
{
BYTE crypt_secret_w_attacker_pub[88]; // same as registry value "1TfXk".
BYTE crypt_secret_w_master_pub[88]; // same as registry value "2YEdLY".
BYTE file_public[32]; // The file_public key.
BYTE salsa20_iv[8]; // salsa20 initialization vector.
DWORD crc32_file_public; // CRC32 hash of file_public.
DWORD et; // The encryption type.
DWORD spsize; // spsize field if applicable.
DWORD encrypted_null; // A NULL value encrypted with salsa20.
} metadata;
/*..SNIP..*/
} rvl_filecrypt_struct, *prvl_filecrypt_struct;
```

Эта подструктура метаданных добавляется в конец каждого зашифрованного файла, используется дешифратором и REvil для проверки того, что файл ранее был зашифрован.

Симметричный ключ шифрования Salsa20 - это хэш SHA-3 общего ключа, полученного из crypt_public ключа жертвы и секрета сгенерированной пары ключей: file_secret.

Чтобы расшифровать файл, должны быть известны как ключ crypt_secret жертвы, так и ключ file_public:

> - Можно расшифровать crypt_secret, зная ключи attacker_secret или master_secret операторов и применяя ту же логику, которая описана в подразделе Шифрование данных.
> - Можно получить ключ для каждого файла file_public из конца зашифрованного файла после проверки его хэша CRC32.
> - Получив как открытый, так и секретный ключи на предыдущих шагах, сгенерируйте общий ключ Curve25519 и хешируйте его с помощью SHA-3.
> - Скопируйте значение вектора инициализации Salsa20 из структуры метаданных в конец файла.
> - Протестируйте расшифровку зашифрованного двойного слова NULL с помощью хэша SHA-3 в качестве ключа.
> - Определите тип шифрования, используемый для правильного дешифрования файла, затем начните дешифрование соответствующим образом.



Следуя этой схеме, злоумышленники не разглашают свои секретные ключи при отправке дешифраторов жертвам. Со своей стороны, злоумышленникам нужно только расшифровать ключ crypt_secret жертвы и отправить обратно дешифратор, встраивающий этот ключ.

Механизм постоянства

До версии 2.1 программа-вымогатель REvil сохраняется на машине, если в поле конфигурации arn установлено значение true. Он записывает свой путь в раздел реестра SOFTWARE для сохранения. Имя значения записи для анализируемого образца — k51299BQXH.

Перед завершением выполнения REvil помечает свой исполняемый файл для удаления при следующей перезагрузке. В результате постоянный путь станет недействительным.

Механизм сохранения был удален из REvil версии 2.1.

Защита

Программа-вымогатель REvil не обеспечивает никакой защиты.

Связь

Программа-вымогатель REvil инициирует обмен данными со своими контроллерами только в том случае, если для поля конфигурации установлено значение true. В этом случае REvil извлекает список контроллеров из строки конфигурации dmn и переходит к построению пользовательского URL-адреса в цикле для каждого контроллера перед инициированием связи.

Каждому контроллеру предшествует строка https: //, за которой следует случайный путь, выбранный из жестко заданных значений, а затем завершается случайным именем файла. Регулярное выражение ниже соответствует всем случайным URL-адресам, созданным программой-вымогателем Revil:

> \/(wp-content|static|content|include|uploads|news|data|admin)\/(images|pictures|image|temp|tmp|graphic|assets|pics|game)\/([a-z]{2}){1,10}\.(jpg|png|gif)

Затем REvil отправляет контроллер POST-запрос, содержащий ключ жертвы, хранящийся в значении реестра QaUXNv2P. После этого ответ контроллера считывается в буфер, который впоследствии игнорируется и освобождается.

Источник: https://intel471.com/blog/revil-ran...analysis-of-a-ransomware-affiliate-operation/
Автор перевода: yashechka
Переведено специально для https://xss.is
CPU register

Пользователь