Hard-Hit HardBit

Анализ версий шифровальщика семейства HardBit и декриптор

Solar JSOC CERT

rt-solar.ru

rt.ru

Ростелеком
Солар

Содержание

1

О команде Solar JSOC CERT

Центр расследования киберинцидентов Solar JSOC CERT начал свою работу в 2017 году. Сегодня подразделение занимается расследованием инцидентов любой сложности, включая наиболее продвинутые атаки от группировок уровня иностранных спецслужб. Так, в 2020 г. эксперты Solar JSOC CERT обнаружили новую группировку TinyScouts, использующую многоступенчатую схему проникновения в инфраструктуру и уникальное вредоносное ПО. В 2021 г. совместно с НКЦКИ выявили и заблокировали серию масштабных атак иностранных хакеров на федеральные органы власти РФ, а также предотвратили попытку ботнета Meris захватить более 45 тыс. устройств. В 2022 г. в ходе расследования одной из кибератак центр выявил новые недостатки безопасности в ПО для защиты рабочих мест – VipNet Client компании «ИнфоТеКС».

Собственная исследовательская лаборатория Solar JSOC CERT ежедневно актуализирует уникальную базу индикаторов и знаний о новых угрозах за счет:

- мониторинга и анализа инфраструктур 280+ клиентов,
- коммерческих подписок,
- информационных обменов,
- развернутой сети сенсоров и ханипотов «Ростелеком-Солар».

Решающую роль в проактивной защите от угроз играет Threat Hunting – одно из направлений работы центра.

Узнать больше о Threat Hunting и Solar JSOC CERT.

2

Вступление

В Solar JSOC CERT обратился заказчик, который был атакован шифровальщиком Hardbit 3.0. Мы провели анализ предоставленного образца и обнаружили возможность расшифровать зашифрованные файлы. В этом отчете мы подробно расскажем, за счет чего и для каких версий HardBit возможна расшифровка (декриптор – в конце отчета), проведем анализ версий шифровальщика, дадим свое предположение, из каких семейств появились первые версии HardBit, а также покажем имеющийся наряду с шифрованием функционал вайпера.

3

Группировка HardBit

Группировка HardBit публично известна с октября 2022 г. Требует выкуп в биткоинах за расшифровку данных. Для связи злоумышленники предлагают электронную почту и мессенджер Tox.

Некоторые источники в марте 2023 г. сообщали, что HardBit использует технику «двойного шифрования» (когда перед шифрованием выполняется выгрузка критичных для жертвы файлов и возможность их опубликования в дальнейшем используется как дополнительный рычаг давления), другие в феврале того же года писали, что подобная техника не используется. Судя по тексту `ransom note`, в его первых версиях не было упоминаний о критичных данных, а со второй версии – появились:

```
Note:
Sensitive data on your system was DOWNLOADED.
If you DON'T WANT your sensitive data to be
PUBLISHED you have to act quickly.
```

У группировки нет DLS-сайта, и мы не нашли информации в открытых источниках о публикации данных от HardBit.

В феврале 2023 HardBit почему-то получила большую огласку в ИБ-сообществе из-за раздела с информацией о киберстраховке в тексте `ransom note`, в котором злоумышленники рекомендуют анонимно сообщить им о наличии и условиях киберстраховки, чтобы они могли использовать эту информацию при общении со страховыми агентами, хотя данный текст был слово в слово скопирован из `ransom note` LockBit 3.0, который появился в июне 2022-го.

4

Анализ версий шифровальщика HardBit

Чтобы понять, какие версии HardBit «подвержены» расшифровке файлов, мы провели анализ всех версий HardBit, которые нам удалось найти.

Сам шифровальщик обычно представляет собой обфусцированный x86 .NET-файл, но в индикаторах компрометации в статьях Varonis и Fortinet был также указан этот файл.

Это 32-разрядный исполняемый файл на Delphi с детектами на вирус Neshta, который известен с 2000-х годов. Оказалось, что данный файл – это просто образец HardBit 2.0 fafbe16c5646bf1776dd3ef62ba905b9b2cb0ee51043859a2f3cdda7dfe20d4c, который был заражен вирусом Neshta. Шифровальщик хранится в `Overlay` файла, 0xa200 байт заголовка которого зашифрованы Neshta. При запуске данного файла шифровальщик копируется в `%temp%\3582-490\<exe-filename>` и запускается. Исходя из вышеописанного, можно смело утверждать, что все образцы HardBit написаны на C#.

Самая первая версия HardBit 1.0 (b919757f99c1668c4b3f5a0c2fd42f918788ceb1d815b64c7d2dda68989ad0e9) была обфусцирована `Ryan-_-Borland_Protector Cracked v1.0`, который мы подробнее опишем далее, так как именно он использовался для обфускации HardBit 2.0 и 3.0. Все последующие версии HardBit 1.0 по неизвестной причине стали обфусцировать с помощью `Crypto Obfuscator For .Net (5.X)`, и у всех таких файлов имелись следующие PDB-пути:

> `C:\Users\`**`Alex`**`\Desktop\Debug\CryptoObfuscator_Output\stub.pdb;`
>
> `C:\Users\`**`Aleksandr`**`\Desktop\Debug\CryptoObfuscator_Output\stub.pdb.`

Забегая вперед, отметим, что в качестве соли в алгоритме шифрования в HardBit 2.0 и выше использовалась строка `"Ivan Medvedev"`. Нам не удалось определить, с чем связана такая любовь к русскоязычным именам. Возможно, с географией разработчиков шифровальщика или это ложные флаги, чтобы сбить с толку исследователей.

При запуске образцов, обфусцированных `Crypto Obfuscator`, всегда в начале выполнялась проверка текущей даты с жестко закодированными датами:

> `01.10.2022 08:46:22 UTC (e0544cf9225461fc1b0e39bbf4f4b1cfcf92e596ae39cfea8ff6933835d5078b)`
>
> `15.10.2022 11:08:03 UTC` (остальные образцы).

В случае запуска после указанных дат возникало необрабатываемое исключение:

> `The assembly is created with an evaluation version of CryptoObfuscator and will stop working on 1-Oct-2022. The evaluation period has expired and the application will now exit.`

Разработчик `Crypto Obfuscator` – индийская компания LogicNP Software. На официальном сайте стоит «Copyright 2020», и большинство версий программ также 2020 г.
Нам неизвестно, существует ли данная компания до сих пор и продолжает ли поддерживать свои продукты.

Возможно, отсутствие поддержки `Crypto Obfuscator` или окончание лицензий на него сподвигло злоумышленников использовать в дальнейшем (вплоть до версии 3.0, а может быть, и далее) в качестве обфускатора `Ryan-_-Borland_Protector Cracked v1.0`.
Этот обфускатор 2017 г. является модификацией популярного обфускатора `ConfuserEx`.

В большинстве образцов не используется control-flow-обфускация. Обфусцированы только названия функций и строки. Также во всех проанализированных нами образцах присутствуют пустые классы со следующими именами других обфускаторов: `NETReactor, DNGuard, Babel, Yano`. Так как классы пустые, то, скорее всего, это сделано для того, чтобы сбить с толку исследователей или программы-деобфускаторы, которые автоматически определяют, чем обфусцирован файл.

Практически во всех версиях шифровальщика `ransom note`, публичный ключ RSA и email хранятся в незашифрованном виде в `Overlay`. Каждое значение разделено разделителем из случайных букв и чисел, похожих на base64, например: `TNhOiIZodsuLpbHuV`. Скорее всего, это сделано для удобства замены этих данных без перекомпиляции шифровальщика (такие образцы имеются в индикаторах компрометации).

Мы хотели бы отдельно отметить одну особенность данного обфускатора. Он изменяет timestamp компиляции файлов на случайный в будущем (2099 год, например). Все образцы HardBit содержат зашифрованные ресурсные сборки (assembly resource dll) или просто ресурсы с различным содержимым (старые `ransom note`, дополнительные файлы для запуска после шифрования и чаще других встречающийся шаблон hta-файла).
Ресурсные сборки также имеют timestamp компиляции, который чаще всего совпадает с timestamp компиляции основного файла, и во всех обнаруженных нами образцах он не был изменен обфускатором. Мы извлекли все подобные timestamp, что позволило понять хронологический порядок появления образцов (рис. 1). Эти данные вместе с дополнительными метаданными представлены в индикаторах компрометации (Прил. 1).

Сами версии шифровальщиков (1.0, 2.0, 3.0 и так далее) идут из метаданных файлов:

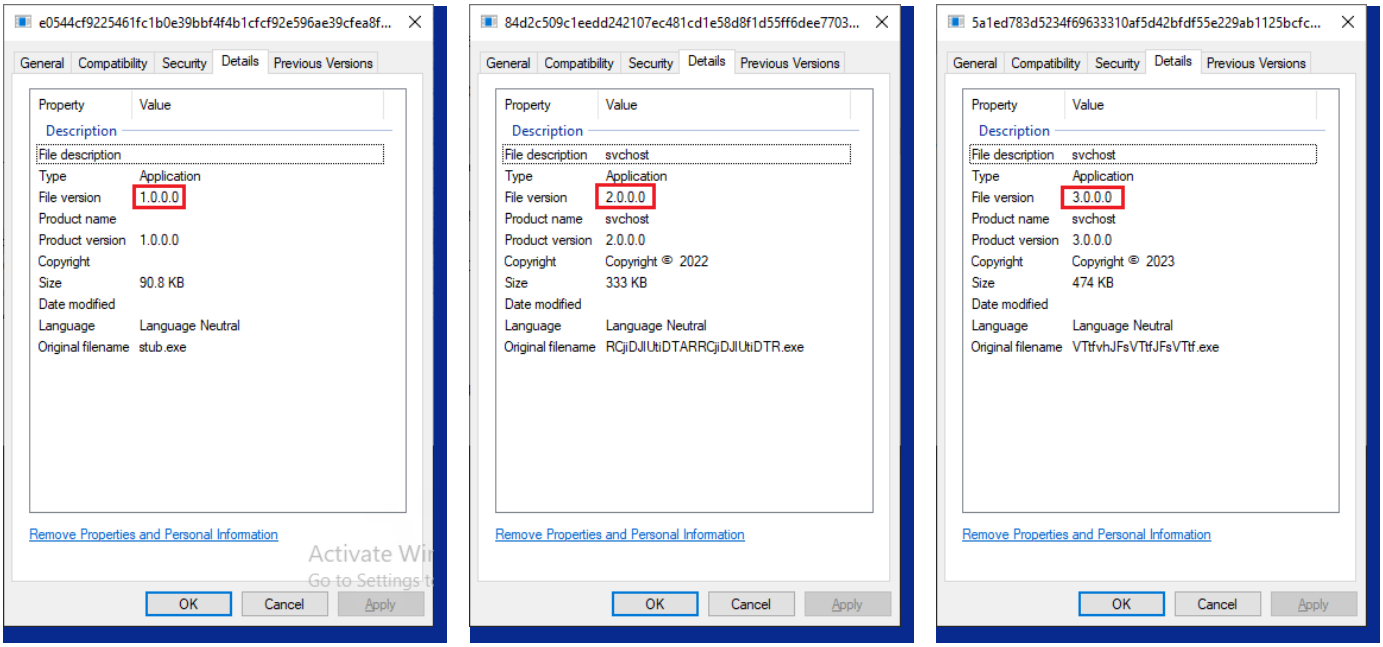


Таблица 1. Версии HardBit

Параметр `Original filename` во всех проанализированных образцах сохранялся. Также версия указывается в расширении зашифрованных файлов:

> `.hardbit, .hardbit2, .hardbit3.`

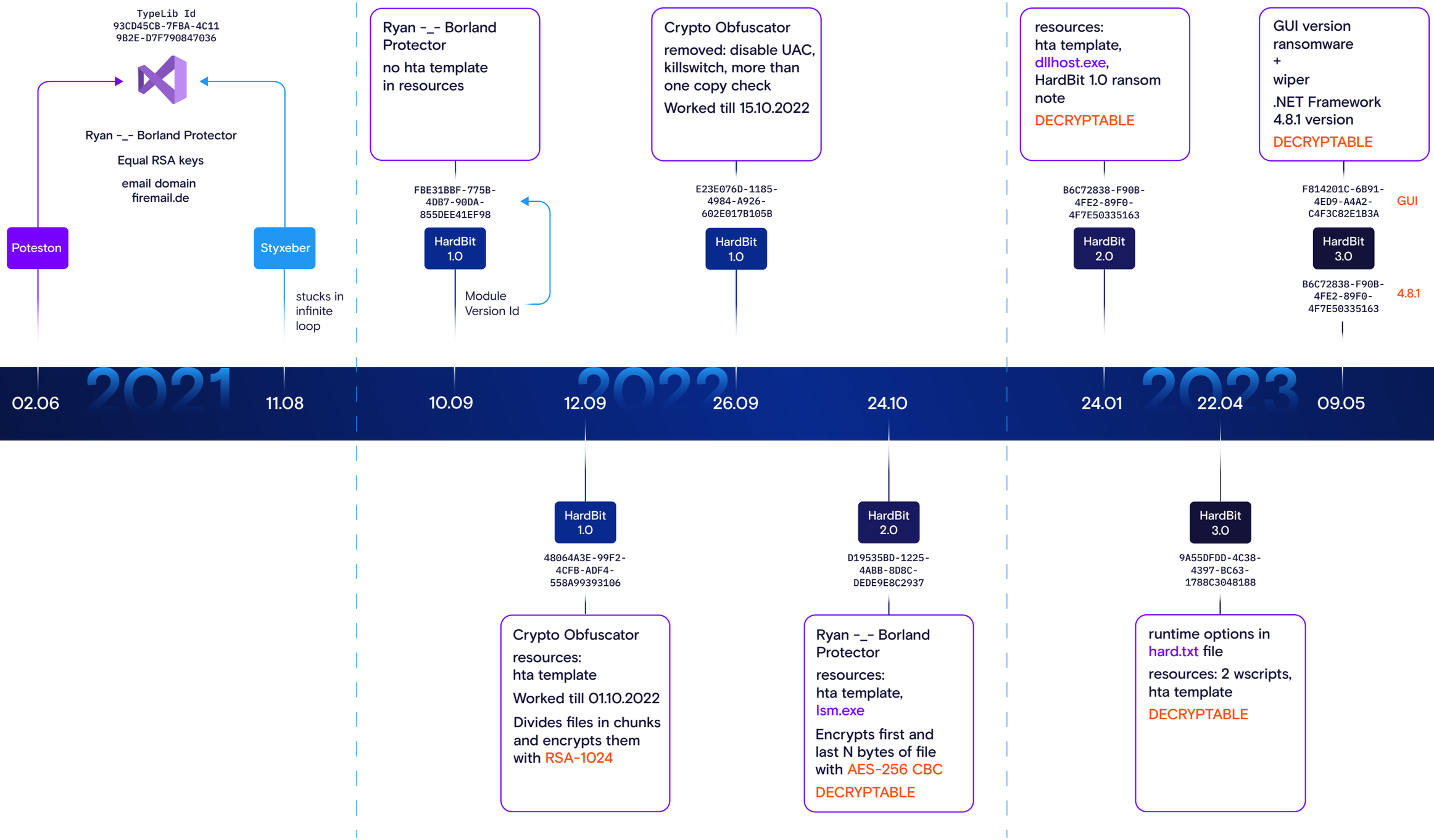


Рисунок 1. Хронология появления образцов HardBit

4.1 HardBit 1.0

Мы нашли 4 образца HardBit 1.0. Очень удобно начинать их анализ на основе метаданных .NET, таких как TypeLib Id и Module Version Id (рис. 2). Очень подробно про это писали на virusbulletin еще в 2015 г. Кратко, TypeLib Id создается Visual Studio и уникален для каждого проекта, Module Version Id генерируется во время сборки для модуля .NET для каждой уникальной сборки.

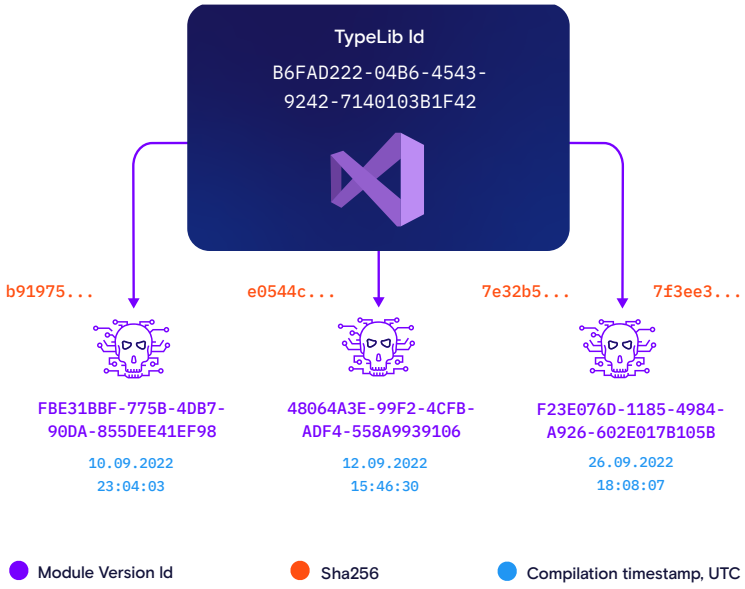


Рисунок 2. Связи HardBit 1.0 на основе метаданных .NET

На рисунке мы видим, что большинство образцов сделаны в одном и том же проекте Visual Studio, а также, что два хеша

> `7f3ee36e9c480457599f0a19eea81a002cce517aca61962214aea165d0699a21` и `7e32b509165bc23ee9d6fa2bed0f049729ac608c6fbdee3daa5e4d6f870ee497`

одинаковы по коду и отличаются только данными в `Overlay`.

Мы не стали изображать на рисунке хеши

> `808d03f47e2ecc4f8f2ef2d03b41c7c191410d162440294a7b493b608fe4cdab` и `c67b531e87130184a92d2479398ace582ea886368b0086562af22b3a9bd3437c`,

так как это образцы HardBit 1.0, которые в ходе расследования

были загружены на VirusTotal исследователями. Они представляют собой дампы образцов

> 7f3ee36e9c480457599f0a19eea81a002cce517aca61962214aea165d0699a21 или 7e32b509165bc23ee9d6fa2bed0f049729ac608c6fbdee3daa5e4d6f870ee497,

так как имеют аналогичные ресурсы, timestamp компиляции и TypeLib Id. Эти образцы исследователей не содержат module .ctor, обработаны de4dot (по именам переименованных методов) и содержат названия методов, которые, скорее всего, были переименованы вручную (например, kill_processes2). Сами файлы не запускаются из-за исключения System.TypeInitializationException.

Возвращаемся к описанию 4 образцов HardBit 1.0. Данные образцы не завершают никаких процессов и служб, не отключают средства защиты. Копируют себя в %appdata%\pp\VGSPWRbBP.exe и этот же путь для обеспечения persistence прописывают в значение "VGSPWRbBP" в реестр-ветку HKLM:SOFTWARE\Microsoft\Windows\CurrentVersion\Run. Перед запуском шифрования удаляют службу «Теневое копирование тома» (VSS или Volume Shadow Copy).

Первые версии

> (хеши b919757f99c1668c4b3f5a0c2fd42f918788ceb1d815b64c7d2dda68989ad0e9 и e0544cf9225461fc1b0e39bbf4f4b1cfcf92e596ae39cfea8ff6933835d5078b)

перед запуском шифрования проверяли наличие файла-индикатора %appdata%\pp\cs.tt (в конце работы шифровальщика создается такой файл со строкой «Done»). Если файл существует, то шифровальщик завершает свою работу. Также проверялось существование файла %appdata\pp\RunAsAdmin.gk. Если файла не существует, то он создается с содержимым "0" и выполняется проверка наличия строки "VGSPWRbBP" в запущенных процессах, затем выполняется отключение UAC через установку значения EnableLua в "0" в реестре и повторный поиск той же строки в процессах. Если строка найдена, шифровальщик завершает свою работу. Если в файле содержится "0", то он перезаписывается на "1" и снова выполняется описанная выше проверка. Если в файле – "1" или указан любой аргумент командной строки, то проверка процессов не выполняется. В следующих версиях все эти проверки убрали.

Шифрование выполняется путем запуска 15 потоков (рис. 3).

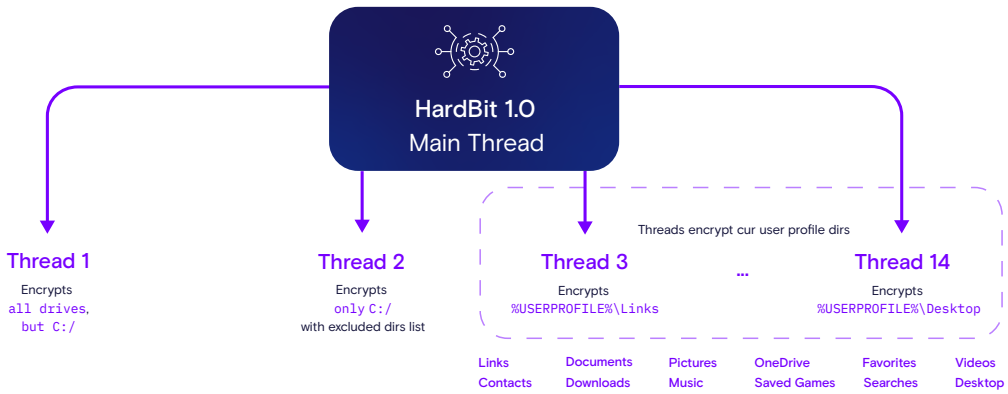


Рисунок 3. Потоки шифрования HardBit 1.0

При шифровании файлов исключаются следующие файлы и расширения:

```
.hardbit
How To Restore Your Files.txt
.Garyk
.dll
.exe
.EXE
.Bin
.lnk
.ini
```

При шифровании диска C:\ дополнительно исключаются каталоги (в потоках шифрования других дисков эти исключения не используются):

```
Windows
Program Files
ProgramData
Temporary Internet Files
PerfLogs
```

Отдельно отметим, что при шифровании диска C:\ шифруются все профили пользователей. Наличие дополнительных потоков шифрования профиля текущего пользователя, скорее всего, вызвано желанием злоумышленников зашифровать профиль текущего пользователя быстрее, чем это сделает поток шифрования диска C:\.

Такими исключениями HardBit старается не нарушить работоспособность ОС после шифрования.

Во всех образцах данной версии RSA-ключ хранится в формате XML-строки, например:

```
<RSAKeyValue>
<Modulus>snMcX74C06yC4w9mvRcV58IH0DXYuT4pJ2TO2agEGJkCbFwBhONmEeFej65W4UfsUG
OFfLQsznyuTidKY3FWezcdQURY1hSWdVm7qQUajT9RIIR0Q9sRGqP66/HZMaQDnysfI5DX+RSRR
nSkrVaE8sedZCK9ieFB995u1ucrrOU=</Modulus> <Exponent>AQAB</Exponent>
</RSAKeyValue>
```

Во всех последующих версиях RSA-ключ хранится в виде BLOB.

Для шифрования файлов перед запуском потоков шифрования генерируется пара сессионных ключей RSA через вызов конструктора `RSACryptoServiceProvider()` без параметров. Сессионный приватный RSA-ключ шифруется встроенным публичным RSA-ключом, кодируется в base64 и является персональным идентификатором `(personal ID)` жертвы в `ransom note`.

При шифровании файл сначала разбивается на части по 0x75 (117) байт, после чего каждая часть шифруется сессионным публичным ключом RSA. Как уже писалось во многих статьях, HardBit перезаписывает файлы шифрованными данными. Шифрование выполняется с конца файла, а в дальнейшем выполняется перезапись файла с нужного смещения. Перед записью шифрованные данные кодируются в `base64` и помещаются в тег `<hardbit>b64_rsa_encrypted_data</hardbit>`.

Если файл без остатка не делится на 0x75 частей, оставшиеся начальные байты не шифруются. Если файл больше 64 KB, то шифруется только 63 999 байт последних байтов файла (0x233 частей по 0x75 байт) (рис. 4).

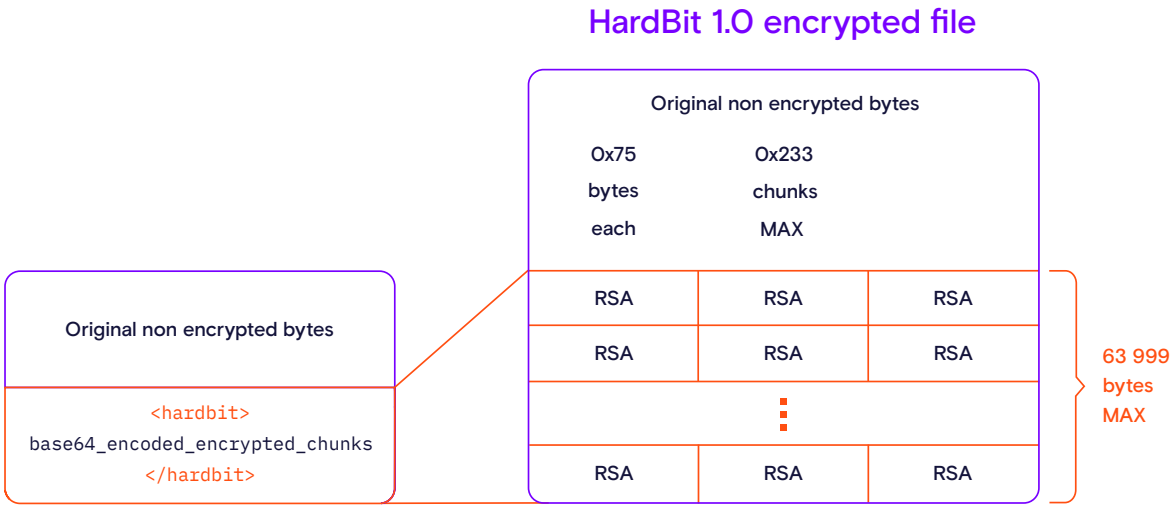


Рисунок 4. Внешний вид файла зашифрованного HardBit 1.0

Отдельно отметим, что имена зашифрованных файлов не меняются. Формат имени зашифрованных файлов:

`filename.ext[id-VictimID].[email].hardbit`, где `filename` – оригинальное имя файла, `ext` – оригинальное расширение файла, `VictimID` – случайные байты из строки `"70F64B87BOPN1XDSEAWSHO7030POGVC4DR5YGFFD6"`. Позиции выбираются с помощью функции `rngcryptoServiceProvider.GetNonZeroBytes(). email` – email злоумышленников, `.hardbit` – расширение, добавляемое шифровальщиком.

Расшифровать файлы, зашифрованные HardBit 1.0, без приватного ключа невозможно.

4.2 HardBit 2.0

Нам удалось найти 7 образцов HardBit 2.0. Приводим результаты анализа на основе метаданных .NET (рис. 5).

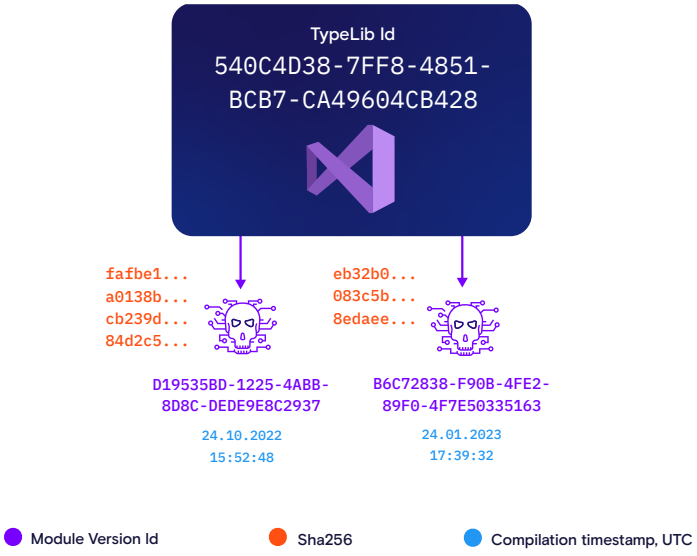


Рисунок 5. Связи HardBit 2.0 на основе метаданных .NET

Как видим, все они были сделаны в одном проекте Visual Studio (отличном от HardBit 1.0) и имеют две «версии» с незначительными отличиями.

> Хеши `fafbe16c5646bf1776dd3ef62ba905b9b2cb0ee510 43859a2f3cdda7dfe20d4c` и другие не имеют Proxy Call Obfuscation и в ресурсах содержат lsm.exe (описан далее) и шаблон hta-файла.

> Хеши `eb32b080535bea5ed7d3c384daf2016048cdc70bf 5f8f25140e7b04e54c5a66b` и другие имеют Proxy Call Obfuscation, другое шифрование строк и в ресурсы добавлены два файла:
>
> - неиспользуемый `ransom note HardBit 1.0` (версию предположили по email-домену `firemail[.]de`);
> - `dllhost.exe` (описан далее).

HardBit 2.0 значительно отличается от первой версии и является фундаментом для третьей версии. Уже было много статей с описанием функционала, поэтому здесь мы остановимся на ключевых изменениях и ранее не описанных возможностях.

Среди ключевых изменений мы выделяем:

- реализован алгоритм подсчета идентификатора жертвы (далее – client_id);
- изменена схема запуска потоков шифрования;
- изменен алгоритм шифрования файлов;
- имена зашифрованных файлов изменяются на случайные;
- изменен persistence: копирует себя в %appdata%\Microsoft\Windows\Start Menu\Programs\Startup\svchost.exe

Эти изменения мы отнесли к ключевым, так как все они продолжат использоваться в HardBit 3.0.

В HardBit 2.0 появился подсчет client_id, который в дальнейшем использовался как пароль для генерации ключа шифрования для алгоритма AES-256 CBC, а также шифровался публичным RSA-ключом, кодировался в base64 и выступал как personal ID в ransom note. Отдельно отметим, что из-за особенностей шифрования RSA personal ID всегда разный, даже при повторном шифровании одного и того же client_id.

Для упрощения приведем псевдокод для генерации client_id:

```
v0 = "SELECT Name, Manufacturer, Version FROM Win32_BaseBoard"
v1 = "SELECT Name, Manufacturer, Version FROM Win32_BIOS"
v2 = "SELECT Name, Manufacturer, ProcessorId from Win32_Processor"
v3 = "SELECT Name, Manufacturer, Model FROM Win32_DiskDrive"
v4 = "SELECT Name, DeviceID, DriverVersion FROM Win32_VideoController"
STR = v0+v1+v2+v3+v4
ProcessorId = Get-WmiObject -Query "SELECT ProcessorId from Win32_
Processor" -Namespace "root\cimv2" | Select-Object -ExpandProperty
ProcessorId -Last 1
Product = "SELECT Product from Win32_BaseBoard"
MacAddress = Get-WmiObject -Query "SELECT MacAddress from Win32_
NetworkAdapterConfiguration where IPEnabled=True" -Namespace "root\cimv2" |
Select-Object -ExpandProperty MACAddress -First 1 | %{$_ -replace ":", ""}
ProcessorId_4b = Get-WmiObject -Query "SELECT ProcessorId from Win32_
Processor" -Namespace "root\cimv2" | Select-Object -ExpandProperty
ProcessorId -Last 1 | % { $_.substring(9,4) }
id = sha1(
STR+
(ProcessorId+Product+MacAddress).toUpper()+
ProcessorId_4b
)
client_id = id+id
```

client_id – это строка (тип String). Для наглядности приводим конкретный пример генерации client_id (рис. 6):

```
id = sha1("Base BoardIntel
CorporationNoneVMW201.00V.20648489.B64.2210180829VMware, Inc.INTEL -
6040000Intel(R) Core(TM) i7-10510U CPU @
1.80GHzGenuineIntel0F8BFBFF000806ECIntel(R) Core(TM) i7-10510U CPU @
1.80GHzGenuineIntel0F8BFBFF000806EC\\.\PHYSICALDRIVE0(Standard disk
drives)VMware Virtual NVMe DiskVMware SVGA
3DVideoController19.17.4.19224AAAF0C80E36F5164EF5F2CF192340080")

md5("0F8BFBFF000806EC440BX Desktop
Reference Platform000C29C01B17")

id = "C4681EDC0C601D10239315018A9A17AED56D2114"

client_id = id+id =
"C4681EDC0C601D10239315018A9A17AED56D2114
C4681EDC0C601D10239315018A9A17AED56D2114"
```

Рисунок 6. Пример подсчета `client_id`

Видно, что `client_id` генерируется на основе, можно сказать, статических параметров конкретного хоста.

Шифрование выполняется путем запуска 15 потоков (рис. 7).

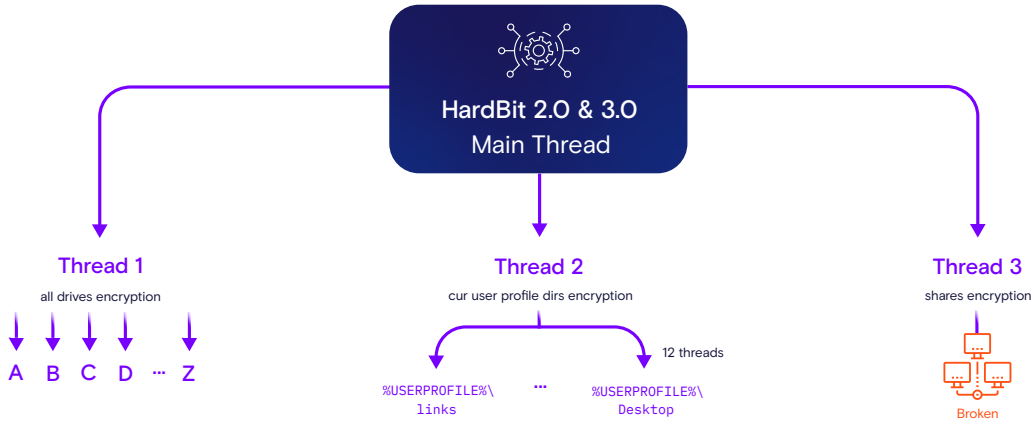


Рисунок 7. Потоки шифрования HardBit 2.0 & 3.0

Как и в HardBit 1.0, исключения на имена файлов и расширения распространяются на все потоки. Исключения на каталоги действуют только для потока шифрования диска C:\. Для HardBit 2.0 не будем приводить конкретные данные, так как в дальнейшем приведем расширенные данные для HardBit 3.0.

Отдельно отметим поток шифрования общих каталогов (shares encryption thread). Для лучшего понимания ситуации приведем его код на C#:

```
IPGlobalProperties ipglobalProperties = IPGlobalProperties.
GetIPGlobalProperties();

string domainName = ipglobalProperties.DomainName;

DirectoryEntry directoryEntry = new DirectoryEntry("WinNT://" +
domainName);

directoryEntry.Children.SchemaFilter.Add("computer)";

foreach (object obj in directoryEntry.Children) {

    DirectoryEntry directoryEntry2 = (DirectoryEntry)obj;

    ransom_obj.encrypt_dir_recursively(directoryEntry2.Name);

}

ransom_obj.encrypt_dir_recursively(string a1_dir_path) {

    ...

    using (Aes aes = new AesManaged()) {

        aes.KeySize = 0x100;

        aes.IV = ransom_obj.IV;

        string[] files = Directory.GetFiles(a1_dir_path);

        string[] directories = Directory.GetDirectories(a1_dir_path);

        int num3 = files.Length - 1;

        for (int i = 0; i <= num3; i++) {

           ransom_obj.encrypt_file(files[i], aes_key);

        }

        int num4 = directories.Length - 1;

        for (int j = 0; j <= num4; j++) {
```

```
            ransom_obj.encrypt_dir_recursively(directories[j]);
            ...
        }
    }
}
```

В самом начале выполняется получение имен всех хостов в домене с использованием legacy - протокола `"WinNT://"`. Этот протокол похож на LDAP и выполняет аналогичные функции. Использовался больше 20 лет назад, поэтому это одна из причин, почему мы решили уделить ему внимание. Поток шифрования общих каталогов на рисунке 7 помечен как «Broken», потому что реально он ничего не шифрует. Вся проблема заключается в том, что полученные имена хостов домена передаются в рекурсивную функцию шифрования каталогов `encrypt_dir_recursively(string a1_dir_path)` в качестве пути (выделено полужирным). Далее в этой функции выполняется попытка получения файлов и каталогов (выделены оранжевым). Это приводит к тому, что имя хоста воспринимается функциями `GetFiles` и `GetDirectories` как относительный путь. После нормализации пути будут выполняться попытки получения файлов и каталогов по текущему пути запущенной сборки, например, `<directory_where_hardbit_2.0_was_executed/<AD_computer_name>`, что приведет к ошибкам. Скорее всего, изначально злоумышленниками планировалось использовать данный поток для шифрования файлов и директорий в общих каталогах.

4.2.1

Файлы lsm.exe и dllhost.exe в ресурсах

Файл `lsm.exe` (ресурс с именем `n8auhs7a` 73b4ab2ae70beb463792Of181ba3f175374209178c86465ca92d333f034ae960) представляет собой дроппер, который запускается после шифрования. В `Overlay` находится зашифрованный `payload`. После запуска создает скрипт `%temp%\is64.bat` и запускает его:

```
@echo off
if exist "%SystemRoot%\Sysnative\" echo:1>"C:\
Users\user\AppData\Local\Temp\is64.txt"
echo:"%SystemRoot%\Sysnative\cmd.exe">C:\Users\
user\AppData\Local\Temp\is64.fil
```

Расшифровывает bat- и exe-файл из `Overlay` в каталог `%temp%\wxy:` Содержимое bat-файла с именем `t<batrandom_5_digits>.bat:`

```
@echo off
set ztmp=C:\Users\user\AppData\Local\Temp\wxy
set MYFILES=C:\Users\user\AppData\Local\Temp\
myfiles
set bfcec=t<exe_random_5_digits>.exe
set cmdline=
SHIFT /0
attrib +h C:\Users\user\AppData\Local\Temp\wxy
```

Если в файле `%temp%\is64.txt` записана 1, то запускает: `C:\Windows\Sysnative\cmd.exe /C C:\Users\user\AppData\Local\Temp\wxy\t<bat_random_5_digits>.bat "C:\Users\user\AppData\Local\Temp\lsm.exe"`.

В данной версии `payload` не расшифровывался, так как на этапе проверки расшифрованные байты не совпадали с жестко закодированными байтами, после чего вылетало окно с ошибкой.

Файл `dllhost.exe` (ресурс с именем `dllhost` efaec6eec913bf80eeb3348e3ee2b9608f546300ff4d1fc5fb9b2d8af2f9eac1) при запуске создает прозрачную `(Form.Opacity = 0)` форму 6x13 пикселей с именем `"audiodg"` без границ и заголовка и скрывает ее из `taskbar.` Текст заголовка: `"Windows Audio Device Graph Isolation "`.

Двигать и изменять размер формы нельзя из-за `Form.FormBorderStyle = None` и `Form.MaximizeBox = false`.

При загрузке формы выполняется код, который **только 24-го числа** создает пустые hta-файлы и непустые `ransom note` в каждом каталоге логических дисков. Даже пытается (на деле это не работает, возникает ошибка) это сделать на всех хостах Active Directory через выше описанную функцию, которая использует протокол WinNT для получения имен хостов в Active Directory. Перед завершением обеспечивает себе persistence через реестр в ключе `HKCU:Software\Microsoft\Windows\CurrentVersion\Run` в значении `"Windows Audio Device Graph Isolation "`, куда прописывает путь, откуда был изначально запущен `dllhost.exe`.

Самое странное, что файл `dllhost.exe` содержит в `Overlay ransom note`, который отличается email-адресами от основного файла, то есть после шифрования данных HardBit 2.0 все hta-файлы будут заменены пустыми, а все `ransom note` с актуальными email будут заменены записками с, вероятно, старыми контактными данными (так как в email используется старый домен `firemail[.]de`, который использовался в HardBit 1.0).

4.2.2

Шифрование файлов HardBit 2.0 & 3.0

Именно в HardBit 2.0 изменился алгоритм шифрования файлов, из-за чего появилась возможность расшифровки зашифрованных файлов. Причем в HardBit 3.0 используется аналогичный алгоритм, поэтому есть возможность расшифровать файлы, зашифрованные как HardBit 2.0, так и HardBit 3.0.

Перед циклом шифрования файлов и каталогов из вышеописанного `client_id` генерируется ключ для алгоритма AES-256 CBC по стандарту PBKDF2 (с помощью C# функции `Rfc2898DeriveBytes(string password,byte[] salt)`). Во всех проанализированных нами образцах HardBit 2.0 и 3.0 в качестве соли и Initialization Vector (IV) использовались одни и те же жестко закодированные значения. В коде на C# это выглядит следующим образом:

```
AesManaged myAes = new AesManaged();
Rfc2898DeriveBytes key = new
Rfc2898DeriveBytes(client_id, Encoding.ASCII.
GetBytes("Ivan Medvedev"));
byte[] IV = new byte[] { 0x5c, 0xd2, 0x23, 0x95,
0xee, 0xef, 0x2a, 0x45, 0x25, 0x47, 0xaa, 0x47,
0x3a, 0xec, 0x45, 0xea };
myAes.Key = key.GetBytes(32);
myAes.IV = IV;
myAes.KeySize = 0x100;
```

Таким образом, можно сделать вывод, что при восстановлении корректного `client_id` можно сгенерировать AES-ключ для расшифровки данных. Перейдем к описанию алгоритма шифрования файлов.

HardBit 2.0 и 3.0 шифруют первые и последние 0x3e800 (256 000) байт файла и дополняют зашифрованные данные следующей структурой метаданных:

Значение	Описание
`char encrypted filename[enc_filename_len]`	Зашифрованное оригинальное имя файла
`char orig_bytes[orig_bytes_len]`	OPTIONAL. Незашифрованные оригинальные байты после смещения `0x3e800`. Присутствует, если размер исходного файла `0x3e800`
`QWORD enc_filename_len`	Длина зашифрованного имени файла
`QWORD END_len`	Размер зашифрованных данных с конца файла
`QWORD orig_file_len`	Исходный размер незашифрованного файла
`QWORD orig_bytes_len`	Количество оригинальных байтов после смещения `0x3e800. [0-16] bytes`
`QWORD BEGIN_len`	Размер зашифрованных данных с начала файла

Таблица 2. Метаданные файла, зашифрованного HardBit 2.0 и 3.0

Отдельно отметим, что `BEGIN_len` и `END_len` – это размеры именно зашифрованных данных, а также, что при шифровании, например, `0x3e800` байт получается `0x3e810` байт зашифрованных данных – это связано с добавлением padding (размер блока AES-256 CBC – 16 байт).

В графическом представлении это выглядит так (рис. 8):

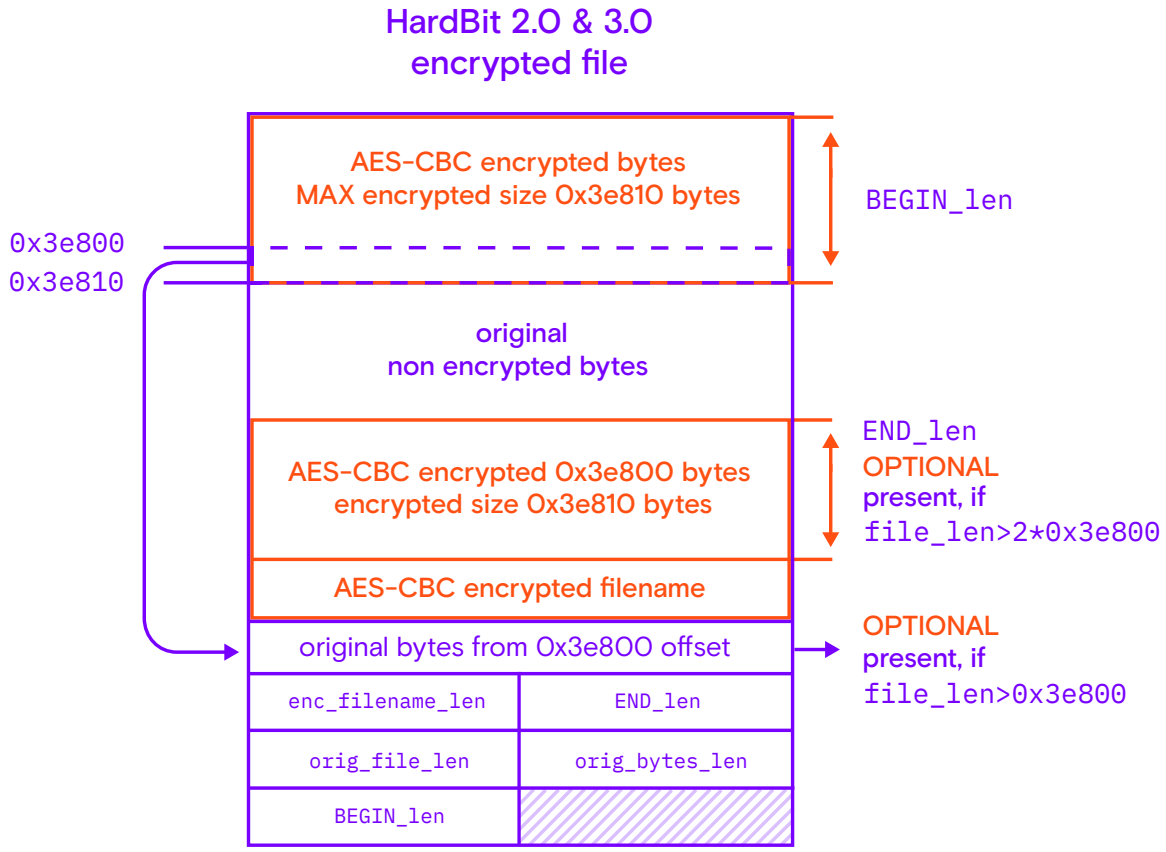


Рисунок 8. Внешний вид файла размером > 512KB, зашифрованного HardBit 2.0 & 3.0

Как видно из вышеописанного, все необходимые для расшифровки данные либо зашифрованы симметричным AES-256 CBC, либо представлены в открытом виде. Злоумышленники шифруют RSA только `client_id`, который помещается в `ransom note`. Совокупность вышерассмотренных факторов позволяет расшифровывать файлы, если получится воссоздать аналогичный `client_id`, который генерировался при запуске шифровальщика.

Перед тем как переходить к описанию самого декриптора, расскажем, какие нововведения появились в HardBit 3.0.

4.3 HardBit 3.0

Нам удалось найти 5 образцов, одного из которых нет на VirusTotal. Приводим результаты анализа на основе метаданных .NET (рис. 9).

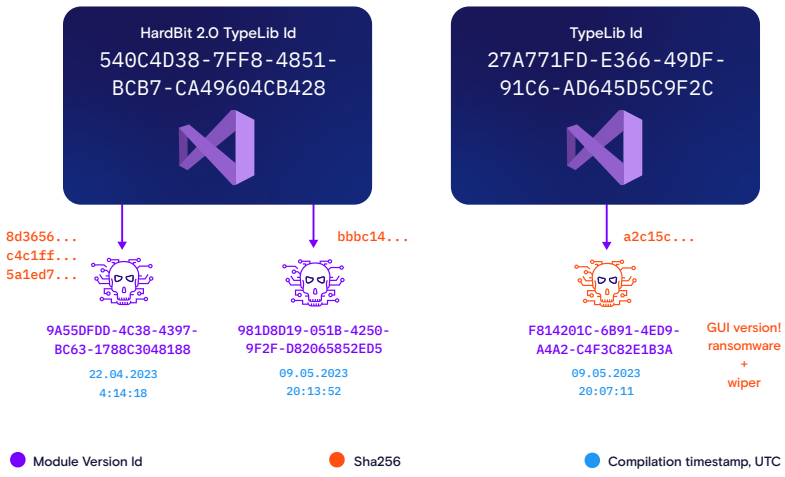


Рисунок 9. Связи HardBit 3.0 на основе метаданных .NET

Версии от TypeLib Id `540C4D38-7FF8-4851-BCB7-CA49604CB428` (который, кстати, совпадает с проектом HardBit 2.0, это подтверждает факт, что версия 3.0 строилась на основе HardBit 2.0) практически не отличаются. Мы нашли разницу только в версии .NET Framework, под которую собирались разные модули. У хеша `bbbc140953e7c6b68ea2abd0a0f9d4d30f6b71b97efaf1efcc915132306d73bb` – 4.8.1, у другой группы хешей – 4.7.2. Сначала опишем общие нововведения HardBit 3.0, далее в отдельном подразделе рассмотрим версию с графическим интерфейсом.

Фундаментом HardBit 3.0, как уже говорилось, является HardBit 2.0. Здесь используются аналогичные алгоритмы для подсчета `client_id` и шифрования файлов, в том числе и такие параметры, как соль и IV. До и после шифрования файлов выполняются действия, аналогичные HardBit 2.0:

```
cmd.exe /C sc delete VSS;
cmd.exe /C wbadmin delete catalog -quiet;
cmd.exe vssadmin delete shadows /all /quiet & wmic
shadowcopy delete;
cmd.exe bcdedit /set {default} bootstatuspolicy
```

> `ignoreallfailures & bcdedit /set {default} recoveryenabled no;`
> отключение Microsoft Defender и всех его механизмов;
> завершение процессов из списка;
> остановка служб из списка. Persistence аналогичен предыдущей версии (через `"%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\svchost.exe"`).

После шифрования наименования зашифрованных логических дисков изменяются на «Locked by Hardbit».

Добавился запуск WScript-скриптов (Прил. 6, 7), которые располагаются в зашифрованной ресурсной сборке вместе с шаблоном hta-файла (рис. 10):

Рисунок 10. Шаблон hta-файла HardBit 3.0

Один WScript удаляет службы по одному списку, другой – останавливает службы по другому списку.

При обычном запуске HardBit 3.0 запускает в том же порядке, такие же и столько же потоков шифрования, как было показано на рисунке 7. Еще раз отметим, что исключения (Прил. 4, 5) на имена файлов и расширения используются всеми потоками. В HardBit 3.0 в поток шифрования диска C:\, помимо использования исключений на каталоги, добавили функцию шифрования профилей пользователей в `C:\Users`, которая также имеет свой список исключений. Добавили файл конфигурации `hard.txt`, который поддерживает различные опции или конкретный путь для шифрования – только один, так как проверяется по регулярному выражению `@"\\|:"` с помощью функции `Regex.IsMatch(File.ReadAllText("hard.txt"), @"\\|:")`. Опции задаются в формате `<option_name>=[false]` или `<option_name>=[true]`, где `<option_name>` – имя опции. Приведем список реализованных опций.

Опция	Описание
-nonshsh	Определяет, будет ли запускаться поток шифрования общих каталогов (shares encryption thread) [true] – не запускать [false] – запустить Default value: [false]
-modefull	Устанавливает полный режим шифрования, при котором последовательно в однопоточном режиме шифруются: - логические диски - каталоги в %USERPROFILE% - общие каталоги (если nonshsh=[false]) [true] – активирует режим [false] – стандартное многопоточное шифрование по схеме HardBit 2.0 Default value: [false]
-modefast	Устанавливает быстрый режим шифрования, при котором меняется очередность запуска потоков: 1) Потоки шифрования данных в %USERPROFILE% 2) Потоки шифрования логических дисков 3) Поток шифрования общих каталогов (если nonshsh=[false]) Напомним, что в обычном режиме сначала запускаются потоки шифрования логических дисков, а потом – потоки шифрования %USERPROFILE% [true] – активирует режим [false] – стандартное многопоточное шифрование по схеме HardBit 2.0 Default value: [false]
-sdel	Отвечает за выполнение самоудаления шифровальщика после окончания своей работы [true] – выполнить самоудаление в конце [false] – не выполнять самоудаление Default value: [false]
Путь до каталога	Выполнить шифрование указанного каталога

Таблица 3. Опции файла конфигурации hard.txt

4.3.1 HardBit 3.0 GUI + Wiper

На рисунке 9 мы показали, что есть образец a2c15c8983710d1647a5eb2ce1ef299fdbcbca0b1b1e0abd8c1b23ffe9f5d9b4, который имеет уникальные TypeLib Id и Module Version Id. Оказалось, что это версия HardBit 3.0 с графическим интерфейсом, который после запуска выглядит следующим образом (рис. 11, 12):

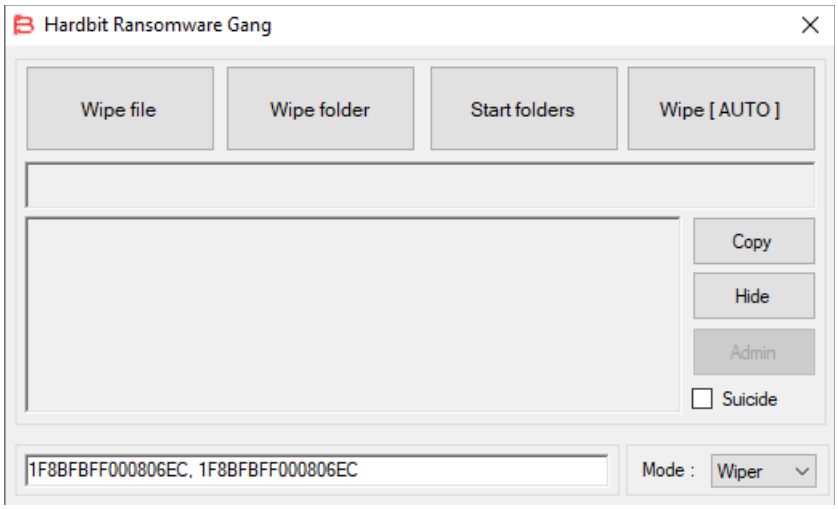


Рисунок 11. Графический интерфейс HardBit 3.0 в режиме вайпера

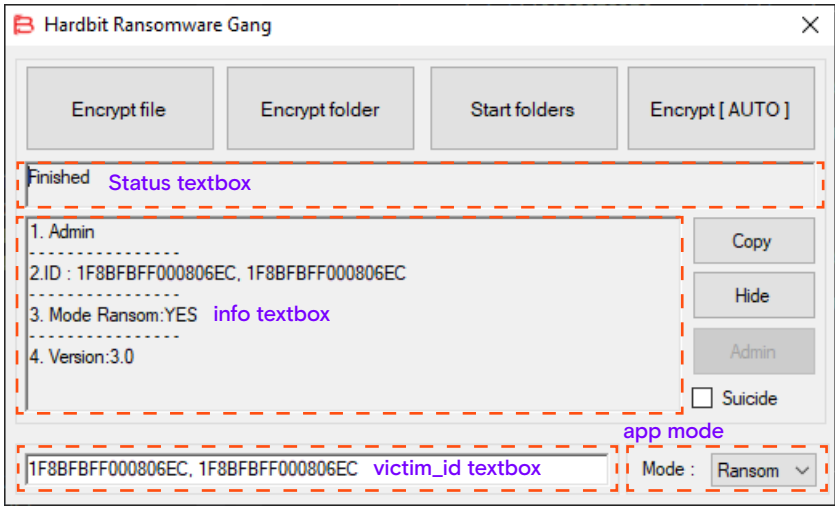


Рисунок 12. Графический интерфейс HardBit 3.0 в режиме шифровальщика

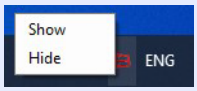
Отметим, что данный образец не выполняет вредоносных действий до нажатия кнопок пользователем. При запуске выполняются только инициализация формы, определение `victim_id`, создание файла-иконки `%appdata%\Microsoft\Windows\Templates\hrdb.ico` и `Default`-значения реестра типа `REG_SZ` в ключе `HKCU\Software\Classes\.hardbit3\DefaultIcon` с этим же путем. В `status textbox` отображаются ход выполнения (имена файлов в процессе работы) и ход завершения действий. В `info textbox` отображается следующая информация:

- наличие прав администратора;
- victim_id;
- режим работы программы;
- версия программы.

С помощью combo box «Mode» можно выбрать режим работы: шифровальщик (Ransom) или вайпер (Wiper).

Отдельно отметим, что для работы в режиме вайпера необходимо, чтобы в каталоге с программой находился файл `hard.txt` с одним из трех публичных ключей RSA-2048 (`p1`, `p2` или `p3` в зашифрованной ресурсной сборке), иначе возникнет окно с текстом ошибки `«You do not have permissions to turn on wipe mode contact us in TOX:<HardBit-TOX-id>».` При активации режима вайпера отображается окно с текстом `«Wiper Mode Activated»`.

Кратко опишем функционал каждого элемента управления HardBit 3.0 GUI в таблице.

Элемент управления	Описание
`"Copy" button`	Копирует в буфер обмена текст из `info textbox`
`"Hide" button`	Скрывает форму программы. Чтобы снова отобразить ее, можно воспользоваться пунктом Show при клике на значок в трее 
`"Admin" button`	Активна, если программа запущена без прав администратора. Выполняет перезапуск программы с помощью `cmd.exe /c powershell stART-PRoceSS <path-to-exe> -veRB rUnAS`

`"Suicide" checkbox`	Работает только для кнопок с `[ AUTO ]`. Если отмечен, то после завершения работы выполняет самоудаление с помощью `cmd.exe /C choice /C Y /N /D Y /T 1 & Del <path-to-exe-file>`
`"Mode" combobox`	Переключатель между режимами работы Ransom и Wiper
`"Encrypt file" &"Wipe file" buttons`	Зашифровать/удалить указанный через диалоговое окно файл
`"Encrypt folder" &"Wipe folder" buttons`	Рекурсивно зашифровать указанный через диалоговое окно каталог, а также записать туда ransom note и hta-файл. В режиме шифрования – обычное рекурсивное удаление каталога

`"Start folders" button`	Шифрует/удаляет файлы и каталоги в текущей директории (из которой была запущена программа). Перед началом работы (в обоих режимах) выполняются следующие действия:
	`cmd.exe /C sc delete VSS`
	`cmd.exe /C wbadmin delete catalog -quiet`
	`cmd.exe vssadmin delete shadows /all /quiet & wmic shadowcopy delete`
	`cmd.exe bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no`
	• отключение Microsoft Defender и всех его механизмов;
	• завершение процессов из списка;
	• остановка служб из списка

`"Encrypt [ AUTO ]" button`	Шифрует все логические диски на хосте. Выполняемые действия аналогичны запуску обычной консольной версии HardBit 2.0 или 3.0 (удаляются теневые копии, обеспечивается persistence, меняются наименования зашифрованных логических дисков и т. д.). Схема запуска потоков аналогична схеме на рисунке 7. По завершении операции в `status textbox` выводится сообщение: `"All Files Are Encrypted"`
`"Wipe [ AUTO ]" button`	Удаляет рекурсивно файлы и каталоги на логических дисках хоста. Выполняется в однопоточном режиме. Последовательность работы и набор исключений полностью аналогичны режиму шифрования с опцией full (отсутствует только функция удаления общих каталогов), только вместо шифрования выполняется удаление. По завершении операции в `status textbox` выводится сообщение: `"Destruction operation was successfully"`

Таблица 4. Функционал элементов управления HardBit 3.0 GUI

Опишем технику удаления файлов вайпером. Перед удалением выполняется перезапись содержимого файла случайными байтами. Если файл < 102 400 байт, то перезапись выполняется случайным байтом, в противном случае перезаписываются первые 102 401 байт массивом случайных байтов. Далее время создания, изменения и доступа изменяются на 26.12.1991 00:00:00 UTC цепочкой следующих вызовов:

```
File.SetLastAccessTime(file_path,dateTime);
File.SetLastWriteTime(file_path,dateTime);
File.SetCreationTimeUtc(file_path,dateTime);
File.SetLastAccessTimeUtc(file_path,dateTime);
File.SetLastWriteTimeUtc(file_path,dateTime);
```

После чего файл удаляется через `File.Delete(file_path)`. Что это за магическая дата 26.12.1991, неизвестно. Возможно, это день рождения разработчика HardBit.

Таким образом, мы видим, что в арсенале HardBit имеется программа с графическим интерфейсом, которая помимо шифрования имеет функционал вайпера. Изначально, пока мы не обнаружили эту версию, казалось, что в HardBit 3.0 не так много нововведений относительно второй версии.

4.4 Образцы до HardBit 1.0

В интервью в октябре 2022 г. HardBit сообщили, что «работают» с ransomware около четырех лет:

> SuspectFile (SF) – The first Hardbit Ransomware file samples have been seen for the first time recently, does this mean that your group is also newly established?
>
> Hardbit Ransomware (Support): We have been working with all kinds of ransomware for almost four years. We put useful features of all ransomware in hardbit
>
> The Hardbit project has been in production for almost 3 years, and we tried to use the safest methods to encrypt files so that both the files remain completely safe and not decrypted, and we officially started working a few months ago.

Во всех образцах HardBit 1.0 в ресурсах мы обнаружили неиспользуемый ресурс `readme`, в котором был текст `ransom note` Poteston ransomware (Прил. 8):

```
All of your files such as Document, photos
,Databases, etc... has been successfully
encrypted! are encrypted by Poteston Ransomware

What guarantees do we give to you?

You can send one of your encrypted file from your
PC and we decrypt it for free.

and files should not contain valuable information
(databases, backups, large excel sheets, etc.).

After payment we will send you the decryption tool
that will decrypt all your files.

Contact us using this email address: recovery_
Potes@firemail.de

Attention!

* Do not rename encrypted files.

* Do not try to decrypt your data using third
party software, it
```

Poteston ransomware появился летом 2021 г.:
https://id-ransomware.blogspot.com/2021/06/poteston-ransomware.html
На VirusTotal нашли два образца с одинаковым TypeLib Id:

sha256	VT First Submission Date, UTC	Compilation Timestamp, UTC	Семейство ransomware
5ad38d579fb249b326a25cffb6f5ffa11b125cda7b6125893432f59a0210	17.06.2021 17:59:32	02.06.2021 12:29:40	Poteston
c94ec73118fc30977608a5768e24e8e07c35c010887d2be922f055184c23	15.09.2021 15:40:28	11.08.2021 14:21:45	Styxeber

Таблица 5. Данные Poteston и Styxeber Ransomware

Имена семейств мы взяли из `ransom note` каждого образца. О семействе Styxeber информации мы не нашли.

Каждый образец обфусцирован тем же обфускатором, что и образцы HardBit – `Ryan-_-Borland_Protector Cracked v1.0`, но с применением control-flow-обфускации. Также имеются AntiVM-проверки:

- строка `"vmware"`в поле `Manufacturer`, приведенном к нижнему регистру, из WMI-запроса `Select * from Win32_ComputerSystem;`
- строка `"VirtualBox"` в поле `Model` из аналогичного WMI-запроса.

Если AntiVM-проверка не пройдена, то вызывается `Thread.Sleep()` на 5 минут и завершение программы.

В обоих образцах используется одинаковый публичный RSA-ключ, который расшифровывается в runtime:

```
BgIAAACkAABSU0ExAAQAAAEAAQCVYVypXZLp5dNFl1P5xvRSr+dGkRjYS3P9bSvudc3RFaS8/nQj28366gab0qxRxXIOzQBQoBhQaUOb8O87ComD9U1anDyI4QJSzUqJy9+Y20CQhA89I9S/NliEoT4Q69nhNDTTbD68jPPVDpA5rpMtjlvCeu5u04IcKmMXuI1J8g==
```

Отметим, что RSA-ключи во всех образцах HardBit не совпадают между собой, поэтому здесь совпадение по ключам и TypeLib Id может указывать на то, что файлы созданы одним и тем же разработчиком (рис. 13).

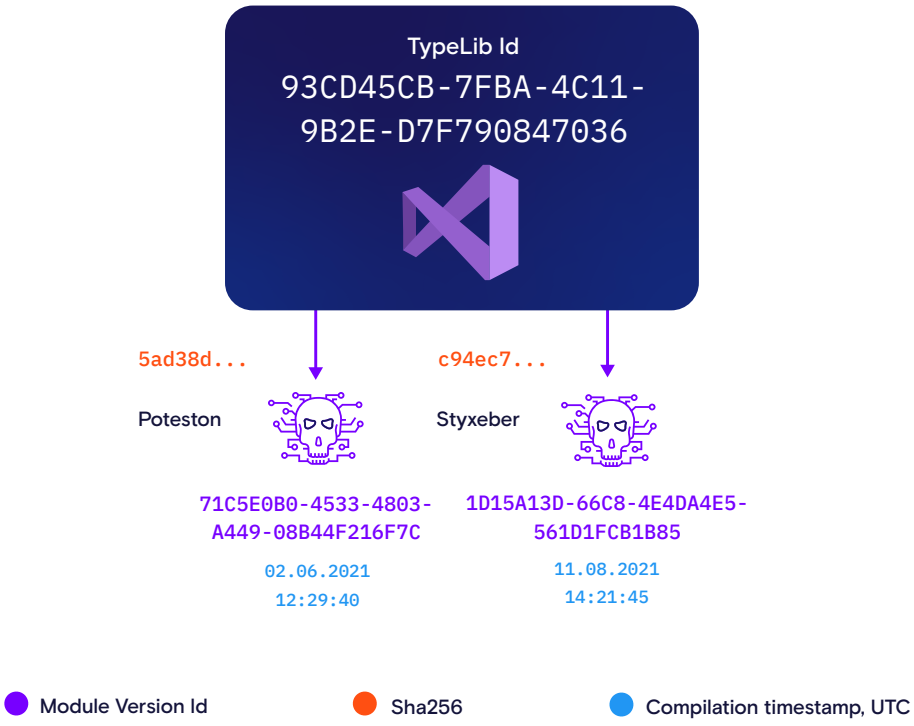


Рисунок 13. Связи первых образцов HardBit на основе метаданных .NET

Мы обнаружили следующие сходства данных образцов с HardBit:

1) Использование обфускатора `Ryan-_-Borland_Protector Cracked v1.0`.
2) Схожесть алгоритма генерации `client_id` с одним из первых образцов HardBit 1.0.

Как уже говорилось выше, во всех образцах HardBit 1.0 `client_id` состоял из 10 символов (параметр функции генерации) и генерировался из строки `"70F64B87BOPN1X DSEAWSHO7030POGVC4DR5YGFFD6"`. Позиции, по которым выбираются байты `client_id`, в свою очередь, выбираются из массива, который генерируется с помощью функции `rngcryptoServiceProvider.GetNonZeroBytes()`. То же самое мы видим в обоих рассматриваемых образцах. Только `client_id` длиной 30 байт, а сами байты выбираются из строки `"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890"`.

3) Шифрование отдельных каталогов в профиле текущего пользователя. В HardBit 1.0 шифруются следующие каталоги в профиле текущего пользователя:

`Links, Contacts, Documents, Downloads, Pictures, Music, OneDrive, Saved Games, Favorites, Searches, Videos, Desktop.`

В рассматриваемых образцах шифруются те же каталоги, только каталог `Contacts` отсутствует.

4) Схожесть в записи `client_id` в `ransom note`.

В HardBit 1.0 и в рассматриваемых образцах `client_id` записывается в `ransom note` в одинаковом формате:

> HardBit 1.0
> `Your ID : <victim_id>`
>
> Poteston
> `Your Personal ID : <b64_rsa_encrypted_client_id>`

Именно так, с пробелом между `"ID"` и `":"`.

Чтобы не было путаницы, подробнее расскажем о `victim_id` и `client_id`. До HardBit 1.0 был только `victim_id`, который генерировался из строки по случайным позициям от функции `rngcryptoServiceProvider.GetNonZeroBytes()`. Он шифровался публичным ключом RSA, так как на его основе генерировался ключ для AES-256 CBC. В HardBit 1.0 использовалось RSA-шифрование с сессионными ключами, поэтому дополнительно к `victim_id` появилась строка `personal id`, куда записывался зашифрованный приватный ключ RSA. В последующих версиях HardBit функцию подсчета `victim_id` заменили на идентификаторы процессоров, разделенные запятой, и добавили функцию генерации `client_id`, на основе которого получался AES-256-CBC-ключ. Этот зашифрованный RSA `client_id` стал помещаться в `personal id`.

5) Использование одного и того же домена для email.

В файле `dllhost.exe` из ресурсов образцов HardBit 2.0 в `ransom note` используется email `manager4hardbit @ firemail[.]de`, в Poteston ransomware `- recovery_Potes @ firemail[.]de`.

Таким образом, можно сделать вывод, что образцы Poteston и Styxeber ransomware, во-первых, связаны между собой и разрабатывались в одном и том же проекте (судя по одинаковым TypeLib Id), во-вторых, скорее всего, являются предшественниками HardBit 1.0 и выступали фундаментом для его разработки.

Для полноты картины приведем краткое описание функционала рассматриваемых образцов. Честно говоря, эти образцы тяжело называть полноценными шифровальщиками. Они похожи на пробы пера, так сказать. Образец Styxeber ничего не шифрует, так как зависает в бесконечном цикле, который отслеживает persistence. Образец Poteston из-за необработанного исключения при шифровании логических дисков шифрует только рассмотренный выше список каталогов в профиле текущего пользователя и в конце переходит в бесконечный цикл отслеживания persistence. Вот эта особенность (то, что образцы не завершаются после запуска) могла использоваться для получения `client_id` или ключа и IV из памяти процесса, с помощью которых зашифрованные файлы можно было бы расшифровать.

Сама процедура рекурсивного шифрования написана очень «криво». Это выражается в том, что каждый файл шифруется столько раз, сколько файлов в текущем каталоге, после чего выполняется шифрование подкаталогов. Другими словами, после шифрования каждого файла в текущем каталоге выполняется рекурсивное шифрование подкаталогов. Это значительно увеличивает общее время шифрования.

Оба образца шифруют файлы полностью с помощью AES-256 CBC. Из шифрования исключаются только файлы с расширением ransomware `(.Poteston, .Styxeber)`.

В отличие от образцов HardBit шифруемые файлы не перезаписываются. Сначала создается зашифрованный файл, после чего старый файл удаляется. К зашифрованному файлу добавляется только соответствующее расширение.

Ключ и IV для AES-256 CBC получаются из строки `victim_id`. Сначала строка переводится в массив байтов, где каждому символу соответствует его код. Далее считается SHA512-хеш от полученного массива байтов. Первые 32 байта – AES-ключ, 16 байт после которого – IV (рис. 14).

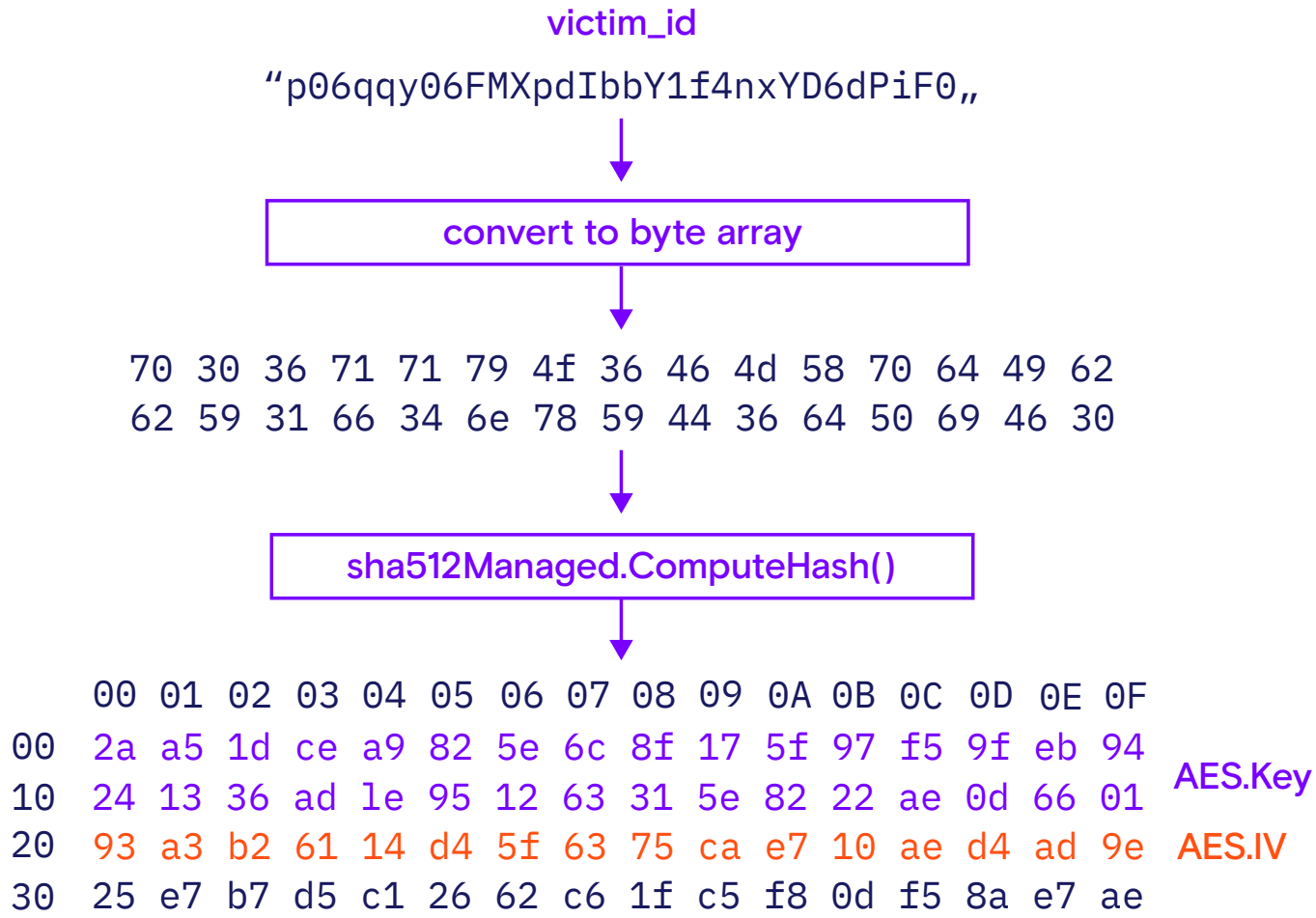


Рисунок 14. Получение ключа и IV из `victim_id` в образцах до HardBit 1.0

Идентификатор жертвы `victim_id` шифруется публичным RSA-ключом и записывается в `ransom note` как `personal ID`.

Перед шифрованием логических дисков и списка каталогов в профиле текущего пользователя выполняется `cmd.exe/c vssadmin.exe delete shadows /all /quiet.`

4.4.1 Poteston ransomware

Для понимания необработанного исключения в Poteston Ransomware приведем его псевдокод:

```
try {

    ...
    encrypt_dir(Path.GetPathRoot(Environment.
    GetFolderPath(Environment.SpecialFolder.
    System,
    (Environment.SpecialFolderOption)(-1))));

}

catch (Exception ex) {

    ...

}

finally {

    encrypt_cur_user_profile_folders()
    set_persistence();

}
```

Метод `Environment.GetFolderPath(Environment.SpecialFolder folder, Environment.SpecialFolderOption option)` почему-то вызывается с `option`, равным -1. Это приводит к появлению необработанного исключения `System.ArgumentOutOfRangeException: Illegal enum value: -1. (Parameter 'option')`, в результате чего выполнение переходит в блок `finally`.

В блоке `finally` выполняется шифрование списка каталогов в профиле текущего пользователя: `Links, Documents, Downloads, Pictures, Music, OneDrive, Saved Games, Favorites, Searches, Videos, Desktop`.

Далее выполняется распространение шифровальщика на съемные носители через создание `autorun.inf` файлов в корне файловой системы (при этом несъемные носители не исключаются), несмотря на то что данный функционал был отключен в Windows с 2009 г. Шифровальщик копирует себя в корень каждого логического диска под именем `SaraJay.exe` и в корне создает файл `autorun.inf` со следующим

содержанием:

```
[autorun]
open=<disk_label>:\SaraJay.exe
shellexecute=<disk_label>:\
```

Файлам присваивается атрибут «Скрытый».

После этого запускается бесконечный цикл, который каждые 5 секунд проверяет наличие исполняемого файла шифровальщика в `%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\`. Если файл отсутствует, то выполняет копирование себя в данный каталог.

4.4.2 Styxeber ransomware

Через пару месяцев после Poteston ransomware появляется образец Styxeber ransomware. После запуска выполняет аналогичные Poteston ransomware AntiVM-проверки, удаляет теневые копии и сразу запускает бесконечный цикл по отслеживанию persistence в каталоге `%appdata%\Microsoft\Windows\Start Menu\Programs\Startup\`. То есть при обычном запуске никакого шифрования выполняться не будет. Возможно, это тестовая версия шифровальщика и название Styxeber тоже тестовое, раз никакой информации о нем нет в открытых источниках.

После этого бесконечного цикла еще имелся функционал, который был похож на Poteston ransomware. Выполнялось шифрование `victim_id` тем же RSA-ключом, что и в Poteston ransomware. Шифровался единственный каталог `Pictures` в профиле текущего пользователя. Далее запускалась аналогичная Poteston Ransomware функция по распространению на съемные носители. Только в этом случае шифровальщик в корень копировал себя под именем `ntmsp.exe`, что практически совпадает с названием ресурса, откуда распространяется вредоносный RTF-документ с `CVE-2017-11882` – но об этом далее. Перед завершением запускался браузер по умолчанию с URL `hxxps://yip[.]su/2wgpJ6`, который перенаправлял на `hxxps://www.google[.]it`.

Судя по информации с VirusTotal, данный образец распространялся через вредоносный RTF-файл, в котором эксплуатировалась уязвимость CVE-2017-11882:

> `hxxp://www.ntmspa[.]com/fra/DoSITE.php?mod=modulistica&snd=file&act=download&id=641;`
>
> `hxxp://www.ntmspa[.]com/ita/DoSITE.php?mod=novita&snd=file&act=download&id=97;`
>
> `hxxp://ntmspa[.]com/ita/DoSITE.php?mod=novita&snd=file&act=download&id=98;`

Ресурс `ntmspa.com` – официальный сайт итальянского завода фитингов и клапанов.

На сайте ntmspa.com в итальянской новостной ленте (hxxp://www.ntmspa[.]com/ita/news.htm) все ссылки ведут на вредоносный RTF-документ 6b756a4cafdbc323ae3240f755a2b5c4cc6528fad214c9ba8302fbc37543faa1, что также видно по одинаковому размеру PDF-файлов (0.56 MB) (рис. 15):

Рисунок 15. Сайт NTM Spa в итальянской редакции с вредоносными ссылками

Внешний и подробный виды RTF-документа (рис. 16, 17):

111111

Рисунок 16. Внешний вид вредоносного RTF-документа

{EMBED Package}{EMBED Equation.3}{EMBED Equation.3}

Рисунок 17. Подробный вид вредоносного RTF-документа

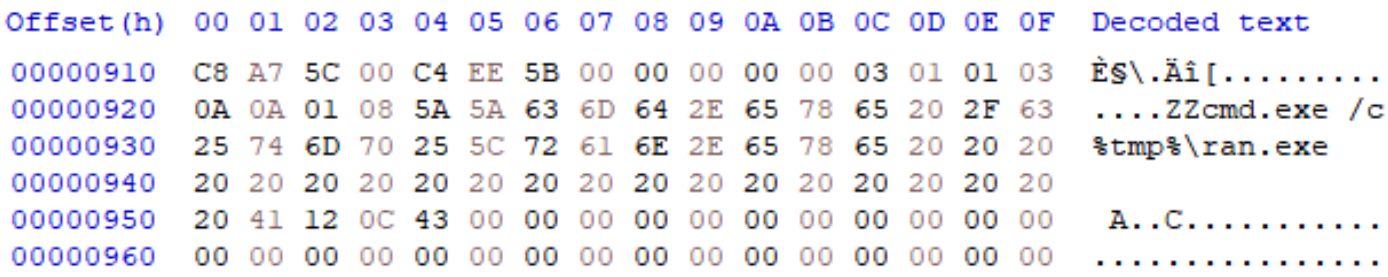
При открытии документа файл `ran.exe`, представляющий собой `Styxeber ransomware (sha256 c94ec73118fc309f77608a5768e24e8be07c35c010887d28be922f055184c231)`, копируется в `%tmp%` и запускается через команду `cmd.exe /c %tmp%\ran.exe`, которая выполняется с помощью API-вызова WinExec.

Первый Equation Object содержит некорректный `payload`, который вызывает исключение в одной из функций обработки шрифтов (рис. 18):

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000CD0   C8 A7 5C 00 C4 EE 5B 00 00 00 00 00 03 01 00 03  È§\.Äî[.........
00000CE0   0A 0A 08 00 01 33 C0 50 8D 44 24 52 50 EB 7F 63  .....3ÀP.D$RPë.c
00000CF0   6D 64 2E 65 78 65 20 2F 63 25 74 6D 70 25 5C 72  md.exe /c%tmp%\r
00000D00   61 6E 2E 65 78 65 20 20 20 20 20 20 20 20 20 20  an.exe
00000D10   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000D20   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000D30   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000D40   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000D50   20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000D60   20 20 20 20 20 20 20 20 20 20 20 20 20 26 90 8B                &.‹
00000D70   44 24 2C 66 2D 51 A8 FF E0 25 00 00 00 00 00 00  D$,f-Q¨ÿà%......
00000D80   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Рисунок 18. Некорректный payload первого OLE-объекта RTF-документа

Второй Equation Object содержит корректный payload, который вызывает `WinExec` (рис. 19):



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000910	C8	A7	5C	00	C4	EE	5B	00	00	00	00	00	03	01	01	03	È§\.Äî[.....
00000920	0A	0A	01	08	5A	5A	63	6D	64	2E	65	78	65	20	2F	63	....ZZcmd.exe /c
00000930	25	74	6D	70	25	5C	72	61	6E	2E	65	78	65	20	20	20	%tmp%\ran.exe
00000940	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000950	20	41	12	0C	43	00	00	00	00	00	00	00	00	00	00	00	A..C.....
00000960	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Рисунок 19. Корректный payload второго OLE-объекта RTF-документа

Скорее всего, сайт `ntmspa.com` был взломан, в результате чего легитимные ссылки заменили на вредоносные. По дизайну и дате обновления последних новостей – 2018 г. для итальянской версии и 2016 г. для всех остальных – создается впечатление, что сайт уже не поддерживается, хотя регистрация домена активна до 2025 года. Скорее всего, планировалось данную «кампанию» нацелить на итальянскую аудиторию, поэтому в коде Styxeber ransomware в конце открывалась ссылка, которая перенаправляла на итальянскую версию Google (google.it). Мы использовали глагол «планировалось», так как напомним, что, во-первых, Styxeber ransomware после запуска попадает в бесконечный цикл отслеживания persistence, во-вторых, даже без бесконечного цикла шифруется только каталог `Pictures` в профиле текущего пользователя, что, по сути, не наносит никакого вреда. `Ransom note` Styxeber ransomware представлен в Приложении 9.

5

Декриптор

Специалисты центра расследования киберинцидентов Solar JSOC CERT написали декриптор для HardBit 2.0 и 3.0.

Его можно загрузить по ссылке:

Скачать декриптор

Декриптор представляет собой консольную программу на .NET 4.8.

Отдельно отметим необходимость использования .NET 4.8 при сборке декриптора, так как в коде генерации `client_id` используется `Encoding.Default`. Если кратко, то на .NET > 4 `Encoding.Default` всегда возвращает `UTF8Encoding`, а не системную кодировку по умолчанию, что может привести к генерации неверного `client_id` и в конечном счете к невозможности расшифровать файлы. Более подробно об этом – на сайте Microsoft.

Детальная инструкция по использованию декриптора располагается там же - на github, чтобы иметь возможность поддерживать ее в актуальном состоянии.

Здесь хотели бы отметить основные требования для успешной расшифровки данных:

1) Для расшифровки необходимо генерировать `client_id`. Для расшифровки данных с хоста N необходимо получить `client_id` именно с этого хоста N. Данный `client_id` может расшифровать файлы только с хоста N. В отдельных уникальных случаях `client_id` может расшифровать файлы и с другого хоста, но только при условии совпадения всех параметров, которые используются для подсчета `client_id`.

2) Если сгенерированный `client_id` не подходит и не расшифровывает файлы, одна из возможных причин – изменение какого-либо параметра, который используется для подсчета `client_id`.

Примеры:

- Отключение/удаление интерфейса, который был активен на момент шифрования.

- Изменение конфигурации виртуальной машины – изменение количества процессоров.
- Подключение/отключение внешнего носителя (жесткого диска).
- Другая причина – проверить расширение зашифрованных файлов. Файлы HardBit 1.0 невозможно расшифровать без приватного RSA-ключа.

6

Заключение

Мы проанализировали различные версии шифровальщика HardBit. Судя по датам компиляции образцов, можно предположить, что у злоумышленников уходит довольно мало времени на выпуск новых версий. Между первой и второй версиями – меньше месяца. Между второй и третьей – 3 месяца. Если сравнить с другими группировками, например с LockBit, то у них между второй и третьей версиями прошел год (LockBit 2.0 – июнь 2021, LockBit 3.0 – июнь 2022). Такой короткий срок выпуска новых версий может быть одной из причин появления возможности расшифровки, так как время на разработку, а также на тестирование сильно ограничено. Скорее всего, после первой версии HardBit хотели увеличить скорость шифрования и перешли на использование симметричного алгоритма шифрования, но не реализовали должным образом все этапы шифрования. Также на примере функции шифрования общих каталогов, которая не выполняет свой функционал, видно, что, вероятно, либо эту функцию не протестировали должным образом, либо у разработчиков не хватило на это времени, и она просто переехала без изменений в HardBit 3.0. С другой стороны, короткий срок выпуска новой версии может привести к тому, что благодаря данному отчету HardBit исправит уязвимости в своем алгоритме шифрования и HardBit 4.0, в появлении которого мы уверены, уже будет невозможно расшифровать без приватных ключей злоумышленников. Предсказать дату релиза новой версии нельзя, но, скорее всего, к концу года можно ожидать новые образцы на VirusTotal.

Также были выявлены предположительные корни HardBit – образцы Poteston и Styxeber ransomware 2021 г., что дополнительно подтверждает факт того, что группировка HardBit давно занимается шифровальщиками, а GUI-версия HardBit 3.0 пополнила арсенал злоумышленников вайпером.

По виктимологии HardBit у нас нет данных. Есть голые факты:

- Большинство статей о HardBit – от зарубежных компаний.
- Ответ HardBit в интервью на вопрос об их целях: «It's just a business».

- Наш случай, когда HardBit атаковали российскую компанию.

Считаем, что на данном этапе развития целями HardBit могут быть предприятия малого и среднего бизнеса в различных странах без каких-либо исключений.

Больше аналитики

Ознакомиться с другими отчетами компании «Ростелеком-Солар» и подписаться на обновления.

Приложение 1. Индикаторы компрометации

sha256	Timestamp, UTC	VT First Submission Date, UTC	Contacts	Original Filename	MVId	TypeLib Id	Comments
Before HardBit 1.0 (Poteston & Styxeber ransomware)							
5ad38d579fb249b3326a25cffb6f5ffea11b125cda7b61205893432f59a02101	02.06.2021 12:29:40	17.06.2021 17:59:32	recovery_Potes@firemail.de	Windows Session Manager.exe	71C5E0B0-4533-4803-A449-08B44F216F7C	93CD45CB-7FBA-4C11-9B2E-D7F790847036	Poteston ransomware. Шифрует только список каталогов (Прил. 3) в профиле текущего пользователя
6b756a4cafdbc323ae3240f755a2b5c4cc6528fad214c9ba8302fbc37543faa1	-	15.09.2021 15:39:45	-	-	-	-	RTF-файл с CVE-2017-11882, запускающий %temp%\ran.exe -> Styxeber ransomware (хеш c94ec7…)
c94ec73118fc309f77608a5768e24e8be07c35c010887d28be922f055184c231	11.08.2021 14:21:45	15.09.2021 15:40:28	Styxeber@firemail.de	svchost.exe	1D15A13D-66C8-4E4D-A4E5-561D1FCB1B85	93CD45CB-7FBA-4C11-9B2E-D7F790847036	Styxeber ransomware. Ничего не шифрует. Зависает в бесконечном цикле
HardBit 1.0							
b919757f99c1668c4b3f5a0c2fd42f918788ceb1d815b64c7d2dda68989ad0e9	10.09.2022 23:04:03	09.03.2023 19:35:05	ineedmyfiles@tutanota.com ineedmyfiles@msgsafe.io	stub.exe	FBE31BBF-775B-4DB7-90DA-855DEE41EF98	B6FAD222-04B6-4543-9242-7140103B1F42	Обфускатор Ryan-_-Borland_Protector Cracked v1.0. В ресурсах нет hta-шаблона
e0544cf9225461fc1b0e39bbf4f4b1cfcf92e596ae39cfea8ff6933835d5078b	12.09.2022 15:46:30	16.09.2022 09:16:31	hardbit@tutanota.com hardbit@msgsafe.io	stub.exe	48064A3E-99F2-4CFB-ADF4-558A99393106	B6FAD222-04B6-4543-9242-7140103B1F42	Crypto Obfuscator. Работал до 01.10.2022 08:46:22 UTC. Появился hta-шаблон. Содержит PDB: C:\Users\Alex\Desktop\Debug\CryptoObfuscator_Output\stub.pdb

sha256	Timestamp, UTC	VT First Submission Date, UTC	Contacts	Original Filename	MVId	TypeLib Id	Comments
				HardBit 1.0			
7f3ee36e9c480457599f0a19eea81a002cce517aca61962214aea165d0699a21	26.09.2022 18:08:07	12.10.2022 06:28:39	boos@keemail.me boos@cyberfear.com	stub.exe	E23E076D-1185-4984-A926-602E017B105B	B6FAD222-04B6-4543-9242-7140103B1F42	Crypto Obfuscator. Работал до 15.10.22 11:08:03 UTC. Убрали проверки на файл-индикатор, отключение UAC и запуск более одной копии. Содержит PDB: C:\Users\Aleksandr\Desktop\Debug\CryptoObfuscator_Output\stub.pdb
808d03f47e2ecc4f8f2ef2d03b41c7c191410d162440294a7b493b608fe4cdab	26.09.2022 18:08:07	31.10.2022 16:20:23	-	stub.exe	63CBAF97-07A2-46FE-AA41-92551D0482D4	B6FAD222-04B6-4543-9242-7140103B1F42	Образцы, с расшифрованными телами методов, без Overlay, обработанные de4dot и с наименованием некоторых функций. Не запускаются из-за ошибки с типами. Содержат ресурсы, аналогичные хешам 7f3ee3…, 7e32b5…
c67b531e87130184a92d2479398ace582ea886368b0086562af22b3a9bd3437c	26.09.2022 18:08:07	11.12.2022 07:15:57	-	stub.exe	3E08CBBB-6ECF-4431-BF42-936901579F7A	B6FAD222-04B6-4543-9242-7140103B1F42	
				HardBit 2.0			
422e0e4e01c826c8a9f31cb3a3b37ba29fb4b4b8c4841e16194258435056d8a3	19.06.1992 22:22:17	22.11.2022 13:46:26	filetest@onionmail.org filetest@decoymail.net	-	-	-	Образец fafbe16c5646bf176dd3ef62ba905bb2cb0ee510438592f3cdda7dfe20d4, зараженный вирусом neshta

sha256	TimestamP UTC	VT First Submission Date, UTC	Contacts	Original Filename	MVId	TypeLib Id	Comments
HardBit 2.0							
73b4ab2ae70beb4637920f181ba3f175374209178c86465ca92d333f034ae960	15.09.2022 13:42:34	22.01.2023 19:24:02	-	-	-	-	Ресурс n8auhs7a (lsm.exe), содержащийся в каждом образце HardBit 2.0
fafbe16c5646bf1776dd3ef62ba905b9b2cb0ee51043859a2f3cdda7dfe20d4c	24.10.2022 15:52:48	22.11.2022 13:55:51	filetest@onionmail.org filetest@decoymail.net	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	D19535BD-1225-4ABB-8D8C-DEDE9E8C2937	540C4D38-7FF8-4851-BCB7-CA49604CB428	Ресурсы: n8auhs7a (lsm.exe) и _00 (hta template). Не имеют Proxy Call Obfuscation
a0138b24593483f50ae7656985b6d6cfe77f7676ba374026199ad49ad26f2992	24.10.2022 15:52:48	20.12.2022 10:40:06	godgood55@tutanota.com alexgod5566@xyzmailpro.com	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	D19535BD-1225-4ABB-8D8C-DEDE9E8C2937	540C4D38-7FF8-4851-BCB7-CA49604CB428	
cb239d641cfa610b1eaf0ecd0f48c42dd147f547b888e4505297c4e9521d8afe	24.10.2022 15:52:48	21.01.2023 18:10:00	filetest@onionmail.org filetest@decoymail.net	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	D19535BD-1225-4ABB-8D8C-DEDE9E8C2937	540C4D38-7FF8-4851-BCB7-CA49604CB428	
efaec6eec913bf80eeb3348e3ee2b9608f546300ff4d1fc5fb9b2d8af2f9eac1	15.01.2023 17:23:29	02.03.2023 01:30:46	manager4hardbit@firemail.de	dllhost.exe	40227D81-D9B6-4B9C-918C-FBA0A8AAF766	37BE3488-C362-412A-AD8A-E9A1667F8593	Ресурс dllhost в хешах с MVId B6C72838-F90B-4FE2-89F0-4F7E50335163. Представляет собой файл со скрытой формой. Закрепляется в run и каждого 24 числа записывает пустые hta-файлы и ransom note (непустые) по всем каталогам логических дисков

sha256	Timestamp, UTC	VT First Submission Date, UTC	Contacts	Original Filename	MVId	TypeLib Id	Comments
			HardBit 2.0				
efaec6eec913bf80eeb3348e3ee2b9608f546300ff4d1fc5fb9b2d8af2f9eac1	24.01.2023 17:39:32	01.03.2023 17:20:24	hardbitman@tutanota.com hardbitman@msgsafe.io	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	B6C72838-F90B-4FE2-89F0-4F7E50335163	540C4D38-7FF8-4851-BCB7-CA49604CB428	Добавлены ресурсы note (старый ransom note, возможно с HardBit 1.0, судя по домену firemail.de, не используется, так как note берется из Overlay) и dllhost (dllhost.exe)
083c5b43df8bee2a6235c3f5038cc9860b4a4bfd1675d367a67fcfff93ccfcfb	24.01.2023 17:39:32	14.04.2023 20:25:06	hardbitfiles@tutanota.com hardbitfiles@msgsafe.io	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	B6C72838-F90B-4FE2-89F0-4F7E50335163	540C4D38-7FF8-4851-BCB7-CA49604CB428	
8edaee2550dde9df1fe2e8c26965be3817f0d66ba13510ac281bfdc8dde1dde7	24.01.2023 17:39:32	15.04.2023 16:35:44	sleepdb@my.com Sleepdb@tutanota.com	RCjiDJlUtiDTARRCjiDJlUtiDTR.exe	B6C72838-F90B-4FE2-89F0-4F7E50335163	540C4D38-7FF8-4851-BCB7-CA49604CB428	Часть строк не зашифрована, остальная часть – с помощью AES. Есть только Proxy Call Obfuscation и изменены имена методов
			HardBit 3.0				
8d365654c14b6192f6f08e1e74ea9ce77f5245347f86faa3c121102faec17f37	22.04.2023 04:14:18	01.05.2023 19:30:43	hardbitfiles@tutanota.com hardbitfiles@msgsafe.io	VTtfvhJFsVTtfJFsVTtf.exe	9A55DFDD-4C38-4397-BC63-1788C3048188	540C4D38-7FF8-4851-BCB7-CA49604CB428	.NET Framework 4.7.2. Ресурсы: 2 wscript-файла для остановки и удаления сервисов (Прил. 6, 7) и шаблон hta-файла. Появился файл hard.txt с параметрами запуска
c4c1ff4bcd2f791729f04afded3ff212c0622c9e31e7b1423af9884f2d529a0c	22.04.2023 04:14:18	09.06.2023 20:19:44	rast@airmail.cc	VTtfvhJFsVTtfJFsVTtf.exe	9A55DFDD-4C38-4397-BC63-1788C3048188	540C4D38-7FF8-4851-BCB7-CA49604CB428	
a2c15c8983710d1647a5eb2ce1ef299fdbcbca0b1b1e0abd8c1b23ffe9f5d9b4	09.05.2023 20:07:11	09.05.2023 23:01:16	helpdec8511@Tutuanota.com decfile855@bingzone.net	VTtfvhJFsVTtfJFsVTtf.exe	9A55DFDD-4C38-4397-BC63-1788C3048188	540C4D38-7FF8-4851-BCB7-CA49604CB428	

sha256	Timestamp, UTC	VT First Submission Date, UTC	Contacts	Original Filename	MVId	TypeLib Id	Comments
			HardBit 3.0				
a2c15c8983710d1647a5eb2ce1ef299fdbcbca0b1b1e0abd8c1b23ffe9f5d9b4	09.05.2023 20:07:11	09.05.2023 23:01:16	helpdec8511@Tutuanota.com decfile855@bingzone.net	VTtfvhJFsVTtfJFsVTtf.exe	F814201C-6B91-4ED9-A4A2-C4F3C82E1B3A	27A771FDE366-49DF-91C6-AD645D5C9F2C	GUI version ransomware + wiper. **Также, скорее всего, имеется опечатка в email-адресе:** tutUanota **вместо** tutanota
bbbc140953e7c6b68ea2abd0a0f9d4d30f6b71b97efaf1efcc915132306d73bb	09.05.2023 20:13:52	-	hisenberg01ger@tutanota.com	VTtfvhJFsVTtfJFsVTtf.exe	981D8D19-051B-4250-9F2F-D82065852ED5	540C4D38-7FF8-4851-BCB7-CA49604CB428	.NET Framework 4.8.1

Приложение 2. HardBit 3.0 ransom note


```
¦¦¦¦¦HARDBIT RANSOMWARE¦¦¦¦¦

Attention!! (Do not scan the files with antivirus in any case. In case of data loss, the consequences are yours) Attention!!

----
what happened?
All your files have been stolen and then encrypted. But don't worry, everything is safe and will be returned to you.
----
How can I get my files back?
You have to pay us to get the files back. We don't have bank or paypal accounts, you only have to pay us via Bitcoin.
----
How can I buy bitcoins?
You can buy bitcoins from all reputable sites in the world and send them to us. Just search how to buy bitcoins on the internet. Our suggestion is these sites.
>>www.binance.com/en<<or>>www.coinbase.com<<or>>localbitcoins.com<<or>>www.bybit.com<<
----
What is your guarantee to restore files?
Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will cooperate with us. Its not in our interests.
To check the ability of returning files, you can send to us any 2 files with SIMPLE extensions(jpg,xls,doc, etc... not databases!) and low sizes(max 1 mb), we will decrypt them and send back to you.
That is our guarantee.
----
How to contact with you?
You can contact us by email:>>>>hisenberg01ger@tutanota.com<<
----
How will the payment process be after payment?
After payment, we will send you the decryption tool along with the guide and we will be with you until the last file is decrypted.
----
What happens if I don't pay you?
If you don't pay us, you will never have access to your files because the private key is only in our hands. This transaction is not important to us,
but it is important to you, because not only do you not have access to your files, but you also lose time. And the more time passes, the more you will lose and
If you do not pay the ransom, we will attack your company again in the future.
----
What are your recommendations?
- Never change the name of the files, if you want to manipulate the files, make sure you make a backup of them. If there is a problem with the files, we are not responsible for it.
- Never work with intermediary companies, because they charge more money from you. For example, if we ask you for 50,000 dollars, they will tell you 55,000 dollars. Don't be afraid of us, just call us.
----
Very important! For those who have cyber insurance against ransomware attacks.
Insurance companies require you to keep your insurance information secret, this is to never pay the maximum amount specified in the contract or to pay nothing at all, disrupting negotiations.
The insurance company will try to derail negotiations in any way they can so that they can later argue that you will be denied coverage because your insurance does not cover the ransom amount.
For example your company is insured for 10 million dollars, while negotiating with your insurance agent about the ransom he will offer us the lowest possible amount, for example 100 thousand dollars,
we will refuse the paltry amount and ask for example the amount of 15 million dollars, the insurance agent will never offer us the top threshold of your insurance of 10 million dollars.
He will do anything to derail negotiations and refuse to pay us out completely and leave you alone with your problem. If you told us anonymously that your company was insured for $10 million and other
important details regarding insurance coverage, we would not demand more than $10 million in correspondence with the insurance agent. That way you would have avoided a leak and decrypted your information.
But since the sneaky insurance agent purposely negotiates so as not to pay for the insurance claim, only the insurance company wins in this situation. To avoid all this and get the money on the insurance,
be sure to inform us anonymously about the availability and terms of insurance coverage, it benefits both you and us, but it does not benefit the insurance company. Poor multimillionaire insurers will not
starve and will not become poorer from the payment of the maximum amount specified in the contract, because everyone knows that the contract is more expensive than money, so let them fulfill the conditions
prescribed in your insurance contract, thanks to our interaction.
-----
```

```
¦¦¦¦¦HARDBIT RANSOMWARE¦¦¦¦¦

Attention!! (Do not scan the files with antivirus in any case. In case of
data loss, the consequences are yours) Attention!!

----

what happened?

All your files have been stolen and then encrypted. But don't worry,
everything is safe and will
be returned to you.

----

How can I get my files back?

You have to pay us to get the files back. We don't have bank or paypal
accounts, you only have
to pay us via Bitcoin.

----

How can I buy bitcoins?
```

You can buy bitcoins from all reputable sites in the world and send them to us. Just search how to buy bitcoins on the internet. Our suggestion is these sites.

>>www.binance.com/en<<or>>www.coinbase.com<<or>>localbitcoins.com<<or>>www.bybit.com<<

What is your guarantee to restore files?

Its just a business. We absolutely do not care about you and your deals, except getting benefits. If we do not do our work and liabilities - nobody will cooperate with us. Its not in our interests.

To check the ability of returning files, you can send to us any 2 files with SIMPLE extensions (jpg,xls,doc, etc... not databases!) and low sizes(max 1 mb), we will decrypt them and send back to you.

That is our guarantee.

How to contact with you?

You can contact us by email:>>>>hisenberg01ger@tutanota.com<<

How will the payment process be after payment?

After payment, we will send you the decryption tool along with the guide and we will be with you until the last file is decrypted.

What happens if I don't pay you?

If you don't pay us, you will never have access to your files because the private key is only in our hands. This transaction is not important to us,

but it is important to you, because not only do you not have access to your files, but you also lose time. And the more time passes, the more you will lose and

If you do not pay the ransom, we will attack your company again in the future.

What are your recommendations?

- Never change the name of the files, if you want to manipulate the files,

make sure you make a backup of them. If there is a problem with the files, we are not responsible for it.

- Never work with intermediary companies, because they charge more money from you. For example, if we ask you for 50,000 dollars, they will tell you

55,000 dollars. Don't be afraid of us, just call us.

Very important! For those who have cyber insurance against ransomware attacks.

Insurance companies require you to keep your insurance information secret, this is to never pay the maximum amount specified in the contract or to pay nothing at all, disrupting negotiations.

The insurance company will try to derail negotiations in any way they can so that they can later argue that you will be denied coverage because your insurance does not cover the ransom amount.

For example your company is insured for 10 million dollars, while negotiating with your insurance agent about the ransom he will offer us the lowest possible amount, for example 100 thousand dollars, we will refuse the paltry amount and ask for example the amount of 15 million dollars, the insurance agent will never offer us the top threshold of your insurance of 10 million dollars.

He will do anything to derail negotiations and refuse to pay us out completely and leave you alone with your problem. If you told us anonymously that your company was insured for $10 million and other important details regarding insurance coverage, we would not demand more than $10 million in correspondence with the insurance agent. That way you would have avoided a leak and decrypted your information.

But since the sneaky insurance agent purposely negotiates so as not to pay for the insurance claim, only the insurance company wins in this situation. To avoid all this and get the money on the insurance, be sure to inform us anonymously about the availability and terms of insurance coverage, it benefits both you and us, but it does not benefit the insurance company. Poor multimillionaire insurers will not starve and will not become poorer from the payment of the maximum amount specified in the contract, because everyone knows that the contract is more expensive than money, so let them fulfill the conditions prescribed in your insurance contract, thanks to our interaction.

Note:

> Sensitive data on your system was DOWNLOADED.
>
> If you DON'T WANT your sensitive data to be PUBLISHED you have to act quickly.
>
> Data includes:
>
> - Employees personal data, CVs, DL, SSN.
>
> - Complete network map including credentials for local and remote services.
>
> - Private financial information including: clients data, bills, budgets, annual reports, bank statements.
>
> - Manufacturing documents including: datagrams, schemas, drawings in solidworks format
>
> - And more...
>
>
> Your ID : <each_physical_processor_ProcessorId_ _divided_by_comma>
>
> Your personal id : <b64_rsa_encrypted_client_id >

Приложение 3. Список каталогов для шифрования в профиле пользователей

```
Desktop
Links
Contacts
Documents
Downloads
Pictures
Music
OneDrive
Saved Games
Searches
Videos
Favorites
```

Приложение 4. HardBit 3.0. Исключения файлов и расширений

```
.386
.DLL
.EXE
.adv
.ani
.bin
.cab
.cmd
.com
.cpl
.cur
.deskthemepack
.diagcab
.diagcfg
.diagpkg
.dll
.drv
.exe
.hardbit3
.hlp
.hta
.icl
.icns
.ico
.ics
```

```
.idx
.key
.lnk
.lock
.mod
.mpa
.msc
.msi
.msp
.msstyles
.msu
.nls
.nomedia
.ocx
.pdb
.prf
.ps1
.rom
.rtp
.scr
.search-ms
.shs
.spl
.sys
.theme
.themepack
.wpx
GDIPFONTCACHEV1.DAT
HARDBIT.jpg
```

```
Help_me_for_Decrypt.hta
How To Restore Your Files.txt
autorun.inf
boot.ini
bootfont.bin
bootsect.bak
d3d9caps.dat
desktop.ini
hard.txt
iconcache.db
ntldr
ntuser.dat
ntuser.ini
thumbs.db
```

Приложение 5. HardBit 3.0. Исключения каталогов при шифровании диска C:\

```
Windows
Program Files
ProgramData
Temporary Internet Files
PerfLogs
Recycle.Bin
boot
program files (x86)
programdata
system volume
tor browser
windows.old
intel
msocache
x64dbg
default
Microsoft
bootmgr
autoexec.bat
Ntuser.ini
Thumbs.db
Config.sys
Boot.ini
Bootsect.bak
AppData
ProgramData
```

`Boot`

`System Volume Information`

`MSOCache`

`Users ->` исключен, так как каталоги пользователей шифруются отдельной функцией после шифрования всего диска `C:\`

`C:\ProgramData`

Приложение 6. HardBit 3.0. WScript для остановки служб

```
Set WshShell = WScript.CreateObject("WScript.Shell")
WshShell.Run "vssadmin Delete Shadows /all /quiet", 0, True
WshShell.Run "net stop SQLAgent$SYSTEM_BGC /y", 0, True
WshShell.Run "net stop ""Sophos Device Control Service"" /y",
0, True
 WshShell.Run "net stop macmnsvc /y", 0, True
 WshShell.Run "net stop SQLAgent$ECWDB2 /y", 0, True
 WshShell.Run "net stop ""Zoolz 2 Service"" /y", 0, True
 WshShell.Run "net stop McTaskManager /y", 0, True
 WshShell.Run "net stop ""Sophos AutoUpdate Service"" /y", 0,
True
 WshShell.Run "net stop ""Sophos System Protection Service"" /y", 0, True
 WshShell.Run "net stop EraserSvc11710 /y", 0, True
 WshShell.Run "net stop PDVFSService /y", 0, True
 WshShell.Run "net stop SQLAgent$PROFXENGAGEMENT /y", 0, True
 WshShell.Run "net stop SAVService /y", 0, True
 WshShell.Run "net stop MSSQLFDLauncher$TPSAMA /y", 0, True
 WshShell.Run "net stop EPSecurityService /y", 0, True
 WshShell.Run "net stop SQLAgent$SOPHOS /y", 0, True
 WshShell.Run "net stop ""Symantec System Recovery"" /y", 0,
True
 WshShell.Run "net stop Antivirus /y", 0, True
 WshShell.Run "net stop SstpSvc /y", 0, True
 WshShell.Run "net stop MSOLAP$SQL_2008 /y", 0, True
 WshShell.Run "net stop TrueKeyServiceHelper /y", 0, True
 WshShell.Run "net stop sacsvr /y", 0, True
```

```
WshShell.Run "net stop VeeamNFSSvc /y", 0, True
WshShell.Run "net stop FA_Scheduler /y", 0, True
WshShell.Run "net stop SAVAdminService /y", 0, True
WshShell.Run "net stop EPUpdateService /y", 0, True
WshShell.Run "net stop VeeamTransportSvc /y", 0, True
WshShell.Run "net stop ""Sophos Health Service"" /y", 0, True
WshShell.Run "net stop bedbg /y", 0, True
WshShell.Run "net stop MSSQLSERVER /y", 0, True
WshShell.Run "net stop KAVFS /y", 0, True
WshShell.Run "net stop Smcinst /y", 0, True
WshShell.Run "net stop MSSQLServerADHelper100 /y", 0, True
WshShell.Run "net stop TmCCSF /y", 0, True
WshShell.Run "net stop wbengine /y", 0, True
WshShell.Run "net stop SQLWriter /y", 0, True
WshShell.Run "net stop MSSQLFDLauncher$TPS /y", 0, True
```

Приложение 7. HardBit 3.0. WScript для удаления служб

```
Set objShell = CreateObject("WScript.Shell")

objShell.Run "cmd /c sc delete vmickvpexchange", 0, True
objShell.Run "cmd /c sc delete vmicguestinterface", 0, True
objShell.Run "cmd /c sc delete vmicshutdown", 0, True
objShell.Run "cmd /c sc delete vmicheartbeat", 0, True
objShell.Run "cmd /c sc delete vmicrdv", 0, True
objShell.Run "cmd /c sc delete storflt", 0, True
objShell.Run "cmd /c sc delete vmictimesync", 0, True
objShell.Run "cmd /c sc delete vmicvss", 0, True
objShell.Run "cmd /c sc delete MSSQLFDLauncher", 0, True
objShell.Run "cmd /c sc delete MSSQLSERVER", 0, True
objShell.Run "cmd /c sc delete 'SQL SERVERAGENT'", 0, True
objShell.Run "cmd /c sc delete SQLBrowser", 0, True
objShell.Run "cmd /c sc delete SQLTELEMETRY", 0, True
objShell.Run "cmd /c sc delete MsDtsServer130", 0, True
objShell.Run "cmd /c sc delete SSISTELEMETRY130", 0, True
objShell.Run "cmd /c sc delete SQLWriter", 0, True
objShell.Run "cmd /c sc delete 'MSSQL$VEEAMSQL2012'", 0, True
objShell.Run "cmd /c sc delete 'SQLAgent$VEEAMSQL2012'", 0,
True
objShell.Run "cmd /c sc delete MSSQL", 0, True
objShell.Run "cmd /c sc delete SQLAgent", 0, True
objShell.Run "cmd /c sc delete MSSQLServerADHelper100", 0,
True
objShell.Run "cmd /c sc delete MSSQLServerOLAPService", 0,
```

```
True
  | objShell.Run "cmd /c sc delete MsDtsServer100", 0, True
  | objShell.Run "cmd /c sc delete ReportServer", 0, True
  | objShell.Run "cmd /c sc delete 'SQLTELEMETRY$HL'", 0, True
  | objShell.Run "cmd /c sc delete TMBMServer", 0, True
  | objShell.Run "cmd /c sc delete 'MSSQL$PROGID'", 0, True
  | objShell.Run "cmd /c sc delete 'MSSQL $WOLTERSKLUWER'", 0,
True
  | objShell.Run "cmd /c sc delete 'SQLAgent$PROGID'", 0, True
  | objShell.Run "cmd /c sc delete 'SQLAgent$WOLTERSKLUWER'", 0,
True
  | objShell.Run "cmd /c sc delete 'MSSQLFDLauncher$OPTIMA'", 0,
True
  | objShell.Run "cmd /c sc delete 'MSS QL$OPTIMA'", 0, True
  | objShell.Run "cmd /c sc delete 'SQLAgent$OPTIMA'", 0, True
  | objShell.Run "cmd /c sc delete 'ReportServer$OPTIMA'", 0,
True
  | objShell.Run "cmd /c sc delete 'msftesql$SQLEXPRESS'", 0,
True
  | objShell.Run "cmd /c sc delete 'postgresql-x64-9.4'", 0, True
```

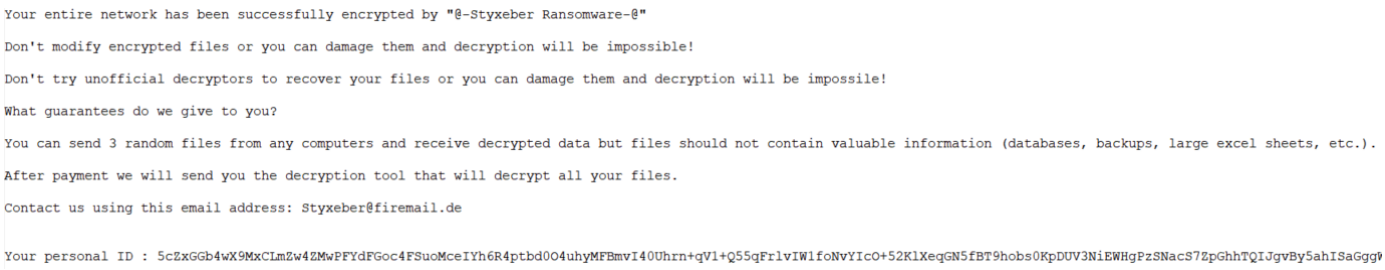
Приложение 8. Poteston ransom note

```
All of your files such as Document, photos ,Databases, etc... has been successfully encrypted! are encrypted by Poteston Ransomware
What guarantees do we give to you?
You can send one of your encrypted file from your PC and we decrypt it for free.
and files should not contain valuable information (databases, backups, large excel sheets, etc.).
After payment we will send you the decryption tool that will decrypt all your files.
Contact us using this email address: recovery_Potes@firemail.de
Attention!
* Do not rename encrypted files.
* Do not try to decrypt your data using third party software, it

Your personal ID : jvcmTByYCglTmRgFOuYqkR22418E/x7fldd2deYFqAUsiQGT95+z2Z+ax2iHHT1cVJmd9kXPe0mCd/N2E7rwV1MOfxtN/UuiYNTCNuqQW1TrWy9+
```

> All of your files such as Document, photos ,Databases, etc... has been successfully encrypted! are encrypted by Poteston Ransomware
>
> What guarantees do we give to you?
>
> You can send one of your encrypted file from your PC and we decrypt it for free.
>
> and files should not contain valuable information (databases, backups, large excel sheets, etc.).
>
> After payment we will send you the decryption tool that will decrypt all your files.
>
> Contact us using this email address: recovery_Potes@firemail.de
>
> Attention!
>
> * Do not rename encrypted files.
>
> * Do not try to decrypt your data using third party software, it
>
> Your personal ID : <b64_rsa_encryped_client_id>

Приложение 9. Styxeber ransom note



```
Your entire network has been successfully encrypted by "@-Styxeber Ransomware-@"
Don't modify encrypted files or you can damage them and decryption will be impossible!
Don't try unofficial decryptors to recover your files or you can damage them and decryption will be impossile!
What guarantees do we give to you?
You can send 3 random files from any computers and receive decrypted data but files should not contain valuable information (databases, backups, large excel sheets, etc.).
After payment we will send you the decryption tool that will decrypt all your files.
Contact us using this email address: Styxeber@firemail.de

Your personal ID : 5cZxGGb4wX9MxCLmZw4ZMwPFYdFGoc4FSuoMceIYh6R4ptbd0O4uhyMFBmvI40Uhrn+qV1+Q55qFrlvIW1foNvYIcO+52KlXeqGN5fBT9hobs0KpDUV3NiEWHgPzSNacS7ZpGhhTQIJgvBy5ahISaGgg9
```

> Your entire network has been successfully encrypted by "@-Styxeber Ransomware-@".
>
> Don't modify encrypted files or you can damage them and decryption will be impossible!
>
> Don't try unofficial decryptors to recover your files or you can damage them and decryption will be impossile!
>
> What guarantees do we give to you?
>
> You can send 3 random files from any computers and receive decrypted data but files should not contain valuable information (databases, backups, large excel sheets, etc.).
>
> After payment we will send you the decryption tool that will decrypt all your files.
>
> Contact us using this email address: Styxeber@firemail.de.
>
> Your personal ID : <b64_rsa_encryped_client_id>.

Больше аналитики

Ознакомиться с другими отчетами компании «Ростелеком-Солар» и подписаться на обновления.