Технология представления результатов сетевого мониторинга в Интернете. Управление системной службой

Терентьев Александр Макарович, кандидат технических наук, PhD
Центральный экономико-математический институт Российской академии наук, г.Москва

Аннотация. *В статье подробно показан пример построения программы управления системной службой web-сервера на языке высокого уровня PowerBASIC. Статья продолжает предыдущий опубликованный материал.*

Ключевые слова: *сетевой мониторинг, информационная безопасность, UDP, web-сервер, системная служба , PowerBASIC.*

Идея круглосуточного низкоуровневого сетевого мониторинга с использованием постоянно работающей наблюдающей станции (НС) в качестве одного из основных средств информационной безопасности была высказана автором достаточно давно [1], сразу после реализации пилотного проекта [2] программных средств.

Возможность дампировать на жёсткий диск НС сетевые пакеты синхронно с их прохождением по сети и затем спокойно разбираться побайтно с их составом сразу же очертила сферы многочисленных возможных применений мониторинга локальной вычислительной сети (ЛВС) [3]. После доведения программных средств до уровня промышленной разработки [4-6], были проведены многочисленные разноплановые эксперименты – выявление «узких мест» в ЛВС ЦЭМИ РАН и выделение её части в специфицированный сегмент сети [7], проведение экспресс-мониторинга в сети малого предприятия с выявлением нарушителя корпоративных правил [8], теоретические исследования структуры ARP-пакетов и сформированных с их девиацией запросов [9], исследование характеристик протоколов обращающихся по сеансам POP3 и FTP пользователей [10], выявление точного подмножества протокола TelNet при обращении к коммутаторам Cisco Catalyst [11-12], проведение суммарного биллинга ЛВС ЦЭМИ РАН за год [13], уточнение деталей протокола выдачи антивирусных обновлений пакета Doctor Web пользователям [14] и другие. Развитие системы сетевого мониторинга в ЦЭМИ РАН помогло решить целый ряд крупных задач, в том числе дефрагментацию ЛВС [7], автоматическое исключение зараженных ПК из ЛВС с помощью управления коммутатором Cisco Catalyst [15] и построение корпоративного варианта технологии снабжения пользователей научных учреждений антивирусными средствами [16-18].

Финальный вариант был описан в [6] и [15]. Особенности мониторинга в условиях, когда центральный хаб заменён на современный коммутатор Cisco Catalyst, приведены в [19]. Некоторые проблемные следствия этого изменения и их решения показаны в [14] и [22].

Дальнейшее развитие сетевого мониторинга связано с проблемой доставки общей информации (главным образом биллинговой [13]) от агрегирующей данные НС мониторной програмы (МП) в Интернет. Первый вариант связи МП↔Сервер с помощью UDP описан в [21]. В этот момент серверную часть исполняло обычное Windows-приложение. В предыдущей работе [26] подробно рассмотрено создание системной службы web-сервера. В данной работе рассмотрено управление такой системной службой.

Мониторная программа, принимающая информацию от НС, исполняется как GUI-приложение на рабочем ПК, соединённым с НС serial-кабелем. Необходимость существования МП вызвана тем, что НС выполнена как приложение в среде MS-DOS и, наблюдая за сетевыми пакетами, сама не способна посылать информацию в локальную сеть, поскольку MS-DOS не является сетевой средой. Таким образом, МП обменивается информацией как с НС, так и с серверной службой.

Как уже было показано [26], преимущество использования системной службы состоит в том, что она начинает работу сразу же при загрузке сервера, до выполнения процедуры Logon. Программа управления системной службой может быть не загружена, что не мешает работе системной службы и исполнению приёма оперативных данных сервером. Таким образом, информационная безопасность сервера уже не зависит от места его установки (при размещении сервера в специально выделенном месте процедура Logon, как правило, не выполняется).

Методы программной реализации самой системной службы и программы управления ею могут быть весьма различны. Современный отечественный стандарт этих методов предполагает использование алгоритмического языка C++. Целью данного цикла работ является, в том числе, желание продемонстрировать менее затратный и более элегантный путь написания подобных программ. Как системная служба, так и программа управления ею выполнены на хорошо известном мировым профессионалам-программистам языке PowerBASIC [23]. Однако, в отличие от системной службы, являющейся консольным приложением, программа управления системной службой является оконным (GUI-) приложением (компилятор PowerBASIC for Windows 9.05). Стоит пояснить, что являясь мощной альтернативой языку среднего уровня C++, PowerBASIC предлагает язык высокого уровня, специфицированный для создания профессиональных Windows-приложений благодаря встроенной поддержке множества обращений к различным библиотекам Windows. Не всегда являясь оптимальным при создании приложений реального времени, он может быть легко дополнен при необходимости процедурами на Макроассемблере [5]. Все версии компиляторов являются платными (в настоящее время порядка $100 за каждую), лицензирование осуществляется вендором исключительно персонально для каждого пользователя.

Приведём «скелет» программы управления службой (рис. 1), осуществляющей отображение полученной информации и контроль исполнения службы. Эта программа представляет собой полноценное GUI-приложение, отображая принимаемые службой по UDP данные и позволяя управлять работой службы. Ряд тривиальных функций (например, запись в лог-файл OutStr), определений и операторов опущен для сокращения текста.

С помощью управляющих кнопок программы контроля можно приостановить и возобновить службу, включить/выключить системный бипер, управляемый службой

(использовалось в основном для контроля взаимосвязи программ), а также выполнить некоторые другие операции управления. Функция ServerThread на рис. 1 исполняется как процесс, параллельный основному, и выполняет, собственно, всю работу (такой метод построения программ типичен для языка PowerBASIC и базируется на основных возможностях системы Windows). Основной процесс только запускает и останавливает тред, а также постоянно следит за веткой реестра.

Как и в любом GUI-приложении, блок CBDialog является собой набор функций, запускаемых системой Windows при каком-либо событии (event), определяемом значением системной переменной CBMsg. Таковым событием может быть инициация или завершение основного диалога, нажатие той или иной управляющей кнопки, изменение режима отображения окна программы или его места на Рабочем столе и другие события (стандартные операторы определения кнопок, как и других элементов окна диалога режима MODELESS, в тексте программы опущены).

```
' TAMOBSSC = Управление Серверной частью Службы. А.Терентьев
#COMPILE EXE
#DIM ALL
#INCLUDE "Win32API.inc"        ' Для большинства системных вызовов
#INCLUDE "TLHELP32.INC"        ' Для определения повторного вызова
.........
$ProgName = "tamobssc"           ' Имя EXE-модуля
$SERVICENAME = "TamObService"    ' Внутреннее имя службы;
                                 ' должно быть уникально на сервере.
$SERVICEFILE = "e:\tamob\tamobss.exe" ' Полный путь к программе службы
FUNCTION Timer_Event () AS LONG ' Вызов каждую 1s
    fTimer=%TRUE: FUNCTION=0: END FUNCTION
FUNCTION SetRegValue(lpKey AS LONG, BYVAL cMainkey AS STRING,_
           BYVAL sKey AS STRING, BYVAL Setting AS STRING) AS LONG
.........
END FUNCTION
.........
CALLBACK FUNCTION CBDialog      ' Обработка диалоговых прерываний
  SELECT CASE CBMSG
    CASE %WM_INITDIALOG           ' Начать диалог
        FUNCTION=1
    CASE %WM_COMMAND              ' Наступило событие (клавиша)
        IF CBCTLMSG = %BN_CLICKED AND _
           CBCTL = %IDC_Start THEN 'Клавиша Запуск службы после останова
               fStart=%TRUE: FUNCTION=1: EXIT SELECT
        END IF
.........                         ' Прочие клавиши
        FUNCTION = 1
    CASE %WM_DESTROY              ' Разрушить диалог
        fDialog=%TRUE: FUNCTION = 1
  END SELECT
END FUNCTION
'============================================================
' Параллельный процесс ServerThread, запускаемый при старте
' - интерпретатор нажатия клавиш и других событий
FUNCTION ServerThread (BYVAL hW AS DWORD) AS LONG
 LOCAL x AS LONG
 WHILE ISFALSE fCloseThr
   IF fStart THEN ' Нажата кнопка Start
     hSCM = OpenSCManager(BYVAL 0, BYVAL 0, %SC_MANAGER_ALL_ACCESS)
     IF hSCM = 0 THEN           ' Не открыт SCM
        dErr = GetLastError()   ' Выяснение причины неоткрытия SCM
        OutStr "OpenSCManager failed. Err = &H" + HEX$(dErr, 8)
     ELSE                       ' SCM нормально открыт
        OutStr "Open SCManager Ok"
        hService = OpenService(hSCM, $SERVICENAME, %SERVICE_START )
        IF hService = 0 THEN       ' Отказ сервиса
          dErr = GetLastError()    ' Выяснение причины отказа
          OUTStr "OpenService failed. Err = &H" + HEX$(dErr, 8)
        ELSE
          fOk = StartService (hService, 0, %NULL) ' Собственно старт
          IF fOk THEN
```

```
            OutStr "Service started Ok"
          ELSE
            dErr=GetLastError()
            OutStr "Service started failed. Err="+HEX$(dErr)
          END IF
          CloseServiceHandle hService ' Обяз. закрыть Handle сервиса
        END IF
        CloseServiceHandle hSCM       ' Обязательно Закрыть Handle SCM
      END IF: fCond=0: fStart=%FALSE
    END IF
.....
    SLEEP 50 ' Это Idle для других программ
    DIALOG GET SIZE hWnd TO x, x        ' Работает ли ещё main dialog?
  WEND  ' ---------------------- Конец основного цикла по fCloseThr
END FUNCTION                  ' Конец ServerThread

FUNCTION PBMAIN () AS LONG  ' Начало основного модуля программы
.....
    DIALOG NEW PIXELS,0, $AppTitle,140,4, 260, 544,_ ' Определение окна
           %WS_POPUP OR %WS_VISIBLE _                ' диалога
           OR %WS_CLIPCHILDREN OR %WS_CAPTION OR %WS_SYSMENU _
           OR %WS_MINIMIZEBOX, 0 TO hDlg    ' Создать dialog box
    DIALOG SHOW MODELESS hDlg CALL CBDialog  ' Немодальный диалог
    iCond=0: fCond=-1: fCloseThr=%FALSE
    THREAD CREATE ServerThread(hDlg) TO hThread ' Параллельный тред
    IF hThread=0 THEN
      MSGBOX "The TAMOBSSC program could not create second thread"+$CRLF _
         +"Windows не позволяет программе TAMOBSSC работать"+$CRLF _
         +" ВНИМАНИЕ! Скорее всего, это действие вируса.", _
           %MB_OK OR %MB_ICONERROR OR %MB_APPLMODAL, $AppTitle
.....
    hTimer = SetTimer(0,0,1000,CODEPTR(Timer_Event)) ' Каждые 1000 mS
    DO                        ' Основной цикл until fDialog is %TRUE
        DIALOG DOEVENTS ' Обязательный оператор в PowerBASIC
        IF ISTRUE fTimer THEN ' Срабатывает каждую секунду
            CurTime  ' Подпрограмма снятия текущего времени и даты
            CONTROL SET TEXT hDlg,%SSTime,cDAT3 ' отразить текущее время
            IF GetRegValue(%HKEY_LOCAL_MACHINE, _
       "SYSTEM\CurrentControlSet\Services\TAMObService", _
       "Signal") = "1" THEN ' Получена очередная порция данных?
               SetRegValue(%HKEY_LOCAL_MACHINE, _
       "SYSTEM\CurrentControlSet\Services\TAMObService", "Signal", "0")
               CALL SetVars ' Основной исполнительный цикл
            END IF
        END IF
    LOOP UNTIL ISTRUE fDialog ' Конец цикла исполнения основного диалога
    fCloseThr = %TRUE         ' Закрытие ServerThread
    DO
        THREAD CLOSE hThread TO Result: SLEEP 50
    LOOP UNTIL ISTRUE Result
    KillTimer 0, hTimer       ' Снять прерывания по таймеру
END FUNCTION
```

Рис. 1. Скелет программы управления системной службой

Процедуры останова на паузу и продолжения системной службы исполняются, как и ранее, через обращения к диспетчеру системных служб Windows – Service Control Manager (SCM).

Использованные в программе серверной службы [26] и программе её управления (Рис. 1) системные вызовы на языке PowerBASIC при компиляции разрешаются с использованием поставляемой с компиляторами библиотеки расширений, содержащей эквивалентные вызовы к стандартным библиотекам Windows (Рис.2). В рисунок включено определение и структура таблицы SERVICE_STATUS обращения к Windows-программе Service Control Manager, использованного много раз в предыдущей статье [26].

```
TYPE SERVICE_STATUS          ' Блок информации для обращений к SCM
  dwServiceType AS DWORD
  dwCurrentState AS DWORD
  dwControlsAccepted AS DWORD
  dwWin32ExitCode AS DWORD
  dwServiceSpecificExitCode AS DWORD
  dwCheckPoint AS DWORD
  dwWaitHint AS DWORD
END TYPE
DECLARE FUNCTION CloseServiceHandle LIB "ADVAPI32.DLL" ALIAS _
    "CloseServiceHandle" (BYVAL hSCObject AS DWORD) AS LONG
DECLARE FUNCTION CreateEvent LIB "KERNEL32.DLL" ALIAS "CreateEventA"_
    (lpEventAttributes AS SECURITY_ATTRIBUTES, BYVAL bManualReset AS_
     LONG, BYVAL bInitialState AS LONG, lpName AS ASCIIZ) AS LONG
DECLARE FUNCTION CreateService LIB "ADVAPI32.DLL" ALIAS "CreateServiceA"_
    (BYVAL hSCManager AS DWORD, lpServiceName AS ASCIIZ, lpDisplayName _
     AS ASCIIZ, BYVAL dwDesiredAccess AS DWORD, BYVAL dwServiceType AS_
     DWORD, BYVAL dwStartType AS DWORD, BYVAL dwErrorControl AS DWORD,_
     lpBinaryPathName AS ASCIIZ, lpLoadOrderGroup AS ASCIIZ, lpdwTagId_
     AS DWORD, lpDependencies AS ASCIIZ, lp AS ASCIIZ, lpPassword AS_
     ASCIIZ) AS LONG
DECLARE FUNCTION GetLastError LIB "KERNEL32.DLL" ALIAS "GetLastError"_
    () AS LONG
DECLARE FUNCTION KillTimer LIB "USER32.DLL" ALIAS "KillTimer" (BYVAL_
     hWnd AS DWORD, BYVAL nIDEvent AS LONG) AS LONG
DECLARE FUNCTION OpenSCManager LIB "ADVAPI32.DLL" ALIAS "OpenSCManagerA"_
    (lpMachineName AS ASCIIZ, lpDatabaseName AS ASCIIZ, _
     BYVAL dwDesiredAccess AS DWORD) AS LONG
DECLARE FUNCTION OpenService LIB "ADVAPI32.DLL" ALIAS "OpenServiceA"_
    (BYVAL hSCManager AS DWORD, lpServiceName AS ASCIIZ, BYVAL _
     dwDesiredAccess AS DWORD) AS LONG
DECLARE FUNCTION RegisterServiceCtrlHandler LIB "ADVAPI32.DLL" ALIAS_
    "RegisterServiceCtrlHandlerA" (lpServiceName AS ASCIIZ, BYVAL _
    lpHandlerProc AS DWORD) AS DWORD
DECLARE FUNCTION SetServiceStatus LIB "ADVAPI32.DLL" ALIAS _
    "SetServiceStatus" (BYVAL hServiceStatus AS DWORD, lpServiceStatus _
    AS SERVICE_STATUS) AS LONG
DECLARE FUNCTION SetTimer LIB "USER32.DLL" ALIAS "SetTimer" (BYVAL _
    hWnd AS DWORD, BYVAL nIDEvent AS LONG, BYVAL uElapse AS DWORD, _
    BYVAL lpTimerFunc AS LONG) AS LONG
DECLARE FUNCTION StartService LIB "ADVAPI32.DLL" ALIAS "StartServiceA"_
   (BYVAL hService AS DWORD, BYVAL dwNumServiceArgs AS DWORD, BYVAL_
   lpServiceArgVectors AS LONG) AS LONG
DECLARE FUNCTION StartServiceCtrlDispatcher LIB "ADVAPI32.DLL" ALIAS_
  "StartServiceCtrlDispatcherA" (lpServiceStartTable AS _
   SERVICE_TABLE_ENTRY) AS LONG
DECLARE FUNCTION RegCreateKeyEx LIB "ADVAPI32.DLL" ALIAS _
   "RegCreateKeyExA" (BYVAL hKey AS DWORD, lpSubKey AS ASCIIZ, BYVAL_
   Reserved AS LONG, lpClass AS ASCIIZ, BYVAL dwOptions AS DWORD, _
   BYVAL samDesired AS DWORD, lpSecurityAttributes AS _
   SECURITY_ATTRIBUTES, phkResult AS DWORD, lpdwDisposition AS DWORD)_
   AS LONG
DECLARE FUNCTION RegSetValueEx LIB "ADVAPI32.DLL" ALIAS _
   "RegSetValueExA"(BYVAL hKey AS DWORD, lpValueName AS ASCIIZ, BYVAL_
   dwReserved AS DWORD, BYVAL dwType AS DWORD, lpData AS ANY, BYVAL _
      cbData AS DWORD) AS LONG
```

Рис. 2. Задействованные системные вызовы в библиотеке PowerBASIC

В процессе своей работы, серверная служба создаёт изменяемый при каждой новой пришедшей порции данных файл на рабочем диске. Программа управления серверной службой, как можно видеть из её основной части, регулярно проверяет состояние некоторой ветви реестра, и, получив в ней значение “Signal”, запускает чтение рабочего файла с диска и смену отображения текущих результатов в окне (соответствующая процедура SetVars ввиду её тривиальности на рис.1 не приведена).

Получение информации в МП от НС идёт ежесекундно. Соединение МП↔Сервер срабатывает каждые 5 секунд. Системная служба обновляет рабочий файл на сервере синхронно с получением очередной информации по UDP. Отображение окна обновляется с интервалом в 6

секунд, в чем можно убедиться непосредственно [24].

Актуальный на момент первой устойчивой версии текст программы зарегистрирован в ФИПС. Часть приведённых программных решений заимствована из многолетних обсуждений на международном форуме программистов по языку PowerBASIC. Конечно, эти решения адаптированы автором под требования конкретной реализации и использованные версии компиляторов.

Можно видеть, что программирование обширного количества системных вызовов к различным библиотекам (KERNEL32.DLL, ADVAPI32.DLL, USER32.DLL и других) легко осуществляется с помощью входящих в поставку компиляторов специальных подключаемых библиотек. Только в основной библиотеке Win32.API содержится более 49000 строк, а всего подключаемых библиотек компилятора более 40. Таким образом, количество строк программы, требующееся для обслуживания системных вызовов, удобно минимизировано по сравнению с другими языками (например, тем же C++), в которых написание подобных сложных функций требует сотен, а то и тысяч программных строк.

Описанным образом реализовано управление системной службой на Антивирусном сервере ЦЭМИ РАН, созданной в целях отображения в Интернете основных агрегирующих данных сетевого мониторинга. Работа программы проверена в средах Windows 2000 Server и Windows Server 2003 R2.

Большинство работ автора, приведённых в списке Литературы, доступны для чтения в разделе «Литература» Антивирусного сайта ЦЭМИ РАН [25].

Литература:

1. Терентьев А.М. Информационная безопасность в крупных локальных сетях. // «Концепции», N1(9)-2002, с.25-30. Свидетельство Роскомпечати 014305.

2. Терентьев А.М. Методы и средства наблюдения загрузки локальных вычислительных сетей на примере ЦЭМИ РАН. / Препринт #WP/2001/110 - М., ЦЭМИ РАН, 2001, - 74с. ISBN 5-8211-0141-7.

3. Терентьев А.М. Задачи полноценного аудита корпоративных сетей // «Концепции», N1(11)-2003, с.94-95. Свидетельство Роскомпечати 014305.

4. Терентьев А.М. Построение и развитие системы сетевого мониторинга. / Развитие и использование средств сетевого мониторинга. Вып.1 Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.5-23. ISBN 5-8211-0317-7.

5. Терентьев А.М. Ускорение форматных преобразований в системах реального времени, реализованных на языке PowerBASIC для i386+. / Развитие и использование средств сетевого мониторинга. Вып.1 Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.24-36. ISBN 5-8211-0317-7.

6. Терентьев А.М. Мониторная программа как средство интеграции данных наблюдающей станции в локальной сети. / Развитие и использование средств сетевого мониторинга. Вып.2. Сб. статей под ред. М.Д.Ильменского. – М.: ЦЭМИ РАН, 2005, с.6-13. ISBN 5-8211-0365-7.

7. Вегнер В.А., Ляпичева Н.Г., Львова А.С., Терентьев А.М. Разработка и реализация типового проекта выделенного сегмента ЛВС на примере ПК административно-финансовой группы ЦЭМИ РАН. / Развитие и использование средств сетевого мониторинга. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.88-101. ISBN 5-8211-0317-7.

8. Терентьев А.М. Опыт сетевого экспресс-мониторинга с помощью переносной наблюдающей станции. / Развитие и использование средств сетевого мониторинга. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.41-46. ISBN 5-8211-0317-7.

9. Терентьев А.М. Об одной побочной возможности использования ARP-пакетов. / Развитие и использование средств сетевого мониторинга. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.37-40. ISBN 5-8211-0317-7.

10. Ляпичева Н.Г., Терентьев А.М. Исследование сетевых сервисов на примере клиентского почтового протокола POP3. / Развитие и использование средств сетевого мониторинга. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.60-74. ISBN 5-8211-0317-7.

11. Терентьев А.М. Возможность полуавтоматического управления сетевыми коммутаторами Cisco Catalyst. / Развитие и использование средств сетевого мониторинга. Вып.2. Сб. статей под ред. М.Д.Ильменского. – М.: ЦЭМИ РАН, 2005, с.14-27. ISBN 5-8211-0365-7.

12. Терентьев А.М. Консоль управления сетевыми коммутаторами Cisco. / Развитие технологий и инструментальных средств информационной безопасности. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2010, с.6-20. ISBN 978-5-8211-0543-1.

13. Ляпичева Н.Г. Выбор Интернет-провайдера на основе измерения трафика / Развитие и использование средств сетевого мониторинга и аудита. Вып. 3. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2006, с.36-49. ISBN 5-8211-0409-2 (978-5-8211-0409-0).

14. Ляпичева Н.Г., Акиншин А.А., Терентьев А.М., Григорьев П.В. Коррекция ошибок HTTP-соединения в локальной сети ЦЭМИ РАН. / Развитие технологий и инструментальных средств информационной безопасности. Вып. 2. Сборник статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2012. – С.49-58.

15. Терентьев А.М. Автоматическая изоляция некорректных объектов КВС по информации сетевого мониторинга. / Развитие технологий и инструментальных средств информационной безопасности. Вып. 3. Сборник статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2012. – С.6-15. ISBN 5-8211-0409-2 (978-5-8211-0409-0).

16. Терентьев А.М. Корпоративный вариант реализации антивирусных пакетов Doctor Web в научных учреждениях: предпосылки. // «Национальные интересы: приоритеты и безопасность». – М.:Издательский дом «Финансы и кредит», N17(206), 2013, с.41-48, ISSN 2073-2872.

17. Терентьев А.М. Корпоративный вариант реализации антивирусных пакетов Doctor Web в научных учреждениях: реализация. // «Национальные интересы: приоритеты и безопасность». – М.:Издательский дом «Финансы и кредит», N19(208), 2013, с.40-45, ISSN 2073-2872.

18. Терентьев А.М. Корпоративный вариант реализации антивирусных пакетов Doctor Web в научных учреждениях:

результаты. // «Национальные интересы: приоритеты и безопасность». – М.:Издательский дом «Финансы и кредит», N20(209), 2013, с.41-46, ISSN 2073-2872.

19. Терентьев А.М., Ляпичева Н.Г., Кочетова Н.А. Мониторинг корпоративной сети ЦЭМИ РАН в условиях использования коммутатора Cisco Catalyst. / Развитие и использование средств сетевого мониторинга и аудита. Вып.1. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2004, с.60-74. ISBN 5-8211-0317-7.

20. Microsoft Windows 2000 Server. Справочник администратора. 2-е изд. / Пер. с англ., - М.: Издательство «СП ЭКОМ», 2003. – 1360 с.: ил. ISBN 5-9570-0002-7.

21. Терентьев А.М., Львова А.С. Адекватное отображение на технологическом www-сервере событий реального времени. / Развитие и использование средств сетевого мониторинга и аудита. Вып.3. Сб. статей под ред. А.М.Терентьева. – М.: ЦЭМИ РАН, 2006, с.50-59. ISBN 5-8211-0409-2.

22. Терентьев А.М. Поддержание доступности HTTP-соединения с помощью периодического пингования. // XXIV Международная научная конференция «Современные концепции научных исследований» – М.: «Евразийское научное объединение», N2(24), 2017. Т.1, с.37-39. ISSN 2411-1899.

23. [Интернет-ресурс] Международный сайт языка программирования PowerBASIC: http://www.powerbasic.com.

24. [Интернет-ресурс] Вызов периодически меняющегося отображения результатов сетевого мониторинга в браузер пользователя: http://av.cemi.rssi.ru/av/r4mon.cgi.

25. [Интернет-ресурс] Антивирусный сайт ЦЭМИ РАН: http://av.cemi.rssi.ru

26. Терентьев А.М. Технология представления результатов сетевого мониторинга в Интернете. Системная служба. // XXVII Международная научная конференция «Стратегии устойчивого развития мировой науки» – М.: «Евразийское научное объединение», N5(27), 2017. Т.1, с.41-46. ISSN 2411-1899.