DOI 10.54398/2074-1707_2022_1_106
УДК 681.3.06

АНАЛИЗ ВОЗМОЖНОСТЕЙ МОНИТОРИНГА ЛОКАЛЬНОЙ ДОМЕННОЙ СЕТИ КЛАССА С СРЕДСТВАМИ MICROSOFT POWERSHELL

Статья поступила в редакцию 18.01.2022, в окончательном варианте – 17.02.2022.

Григорьев Роман Игоревич, Уральский технический институт связи и информатики (филиал) ФГБОУ ВО «Сибирский государственный университет телекоммуникаций и информатики», 620109, Российская Федерация, г. Екатеринбург, ул. Репина, 15,
студент, ORCID: 0000-0002-2835-3807, e-mail: c.terner@mail.ru
Осипова Ирина Александровна, Уральский технический институт связи и информатики (филиал) ФГБОУ ВО «Сибирский государственный университет телекоммуникаций и информатики», 620109, Российская Федерация, г. Екатеринбург, ул. Репина, 15,
кандидат технических наук, доцент, ORCID: 0000-0002-3823-1141, e-mail: minesur@mail.ru

В статье рассмотрены этапы разработки пользовательского модуля Powershell для мониторинга сети класса С. Проанализированы текущие проблемы в локальной сети библиотеки им. В.Г. Белинского. Сформированы технические задачи системного администратора по их устранению. Указаны особенности средств автоматизации Powershell. Проанализированы предварительные действия для развертывания Powershell. Предложена концепция процесса сетевого мониторинга. На основе задач разработаны некоторые функции модуля Powershell и рассмотрен принцип проектирования других. Протестированы функции обнаружения устройств в сети и формирования отчета о рабочих станциях под управлением Windows в формате HTML. Рассмотрен репозиторий пользовательских модулей PowershellGallery. Показана структура работы модуля и порядок действий при его использовании.

Ключевые слова: Powershell, средство автоматизации, технология WMI, локальная сеть, отчет о компьютере, формат отчета HTML, модуль Powershell, функции Powershell, сбор информации о компьютерах, обнаружение устройств в сети, PowershellGallery, система мониторинга

ANALYSIS OF FEATURES OF MONITORING OF LOCAL DOMAIN NETWOK CLASS C BY USING THE TOOL MICROSOFT POWERSHELL

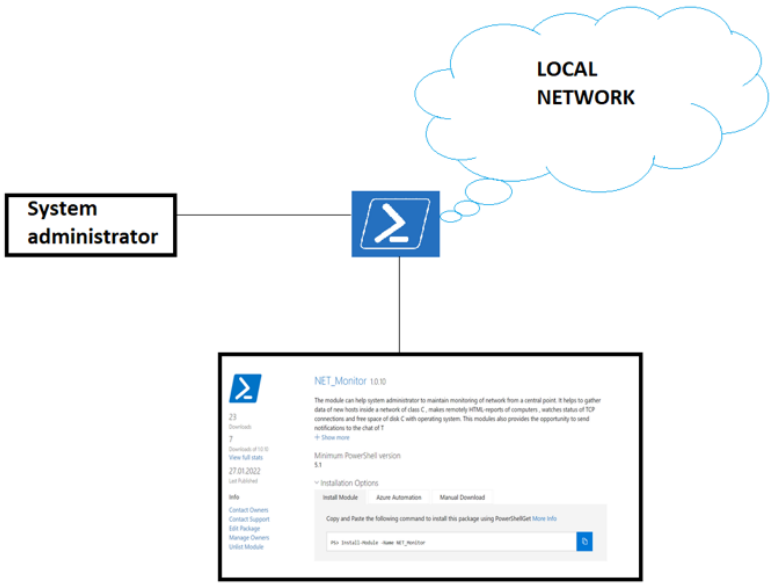
The article was received by the editorial board on 18.01.2022, in the final version – 17.02.2022.

Grigoriev Roman I., Ural Technical University of Connection and Informatics, 15 Repin St., Yekaterinburg, 620109, Russian Federation,
student, ORCID: 0000-0002-2835-3807 e-mail: c.terner@mail.ru
Osipova Irina A., The Ural Technical University of Connection and Informatics, 15 Repin St., Yekaterinburg, 620109, Russian Federation,
Cand. Sci. (Engineering), Assistant Professor, ORCID: 0000-0002-3823-1141, e-mail: minesur @mail.ru

The article shows stages of development of the custom Powershell module which is designed for monitoring networks. It analyzed current problems of local network of SOUNB named after V.G. Belinskiy. The technical tasks for system administrator were also formedto fix current problems. The article considers features of automation tool Powershell. It analyzed preparatory actions for deployment of Powershell. The conception of process of monitoring of network were offered to consideration. Depends on tasks some functions of the module were constructed, and schematics of others were considered. The functions of detection of other devices and of making HTML reports of working stations were tested. The article considers the repository of custom modules PowershellGallery. The structure of work of the module and actions of its deployment were demonstrated.

Keywords: Powershell, automation tool, WMI technology, local network, report of computer, format of report HTML, Powershell module, Powershell functions, gathering data of computers, detection of devices, PowershellGallery, system of monitoring

Графическая аннотация (Graphical annotation)



Введение. В данной работе исследуется возможность создания персонализированных инструментов мониторинга локальной сети класса С для небольших предприятий без использования стороннего программного обеспечения и комплексных систем мониторинга. Улучшенные функции Microsoft Powershell с возможностями командлетов группируются в отдельный модуль, который загружается в центральный репозиторий Powershell Gallery. Системный администратор сможет самостоятельно определять, какой функционал ему необходим для поддержания сетевой инфраструктуры. В дальнейшем, при помощи одной команды Install-Module, такой модуль можно установить на любой компьютер в локальной сети и поддерживать централизованное управление всеми хостами из одной точки сети. Потенциал применения таких функций позволяет расширить функционал библиотек Microsoft Powershell и, при использовании платных графических средств Powershell, даже позволит реализовать собственную систему мониторинга, скомпилированную в exe-файл при помощи модуля Powershell ps2exe.

Процесс поддержания работоспособности локальной вычислительной сети на предприятии является основной ответственностью системного администратора. Для того чтобы эффективно устранять возникающие неполадки, специалист должен периодически проводить мониторинг сети с целью обновления списка устройств в сети и информации о каждом из них. Наиболее используемыми форматами для обработки информации являются текстовые и табличные, в то время как для взаимодействия с человеком предпочтительно использование HTML-формата гиперссылок.

На настоящий момент времени существует множество систем мониторинга – платных и с открытым кодом. В работе Лукаса Макуры “Multi-riteria Analysis and Prediction of Network Incidents Using Monitoring System” и Эмануэля Згардея “Improving IT Infrastructure Management Using Nagios Open Source Package” были рассмотрены наиболее оптимизированные решения мониторинга с открытым исходным кодом – Nagios, Zabbix [1, 2]. Среди платных продуктов можно выделить Network Olympus, Paessler PRTG Network Monitor, 10-Страйк Мониторинг и др.

При рассмотрении тенденций обеих этих групп у каждой из них можно выделить ряд характерных признаков, которые показывают определенные достоинства и недостатки. Системы мониторинга с открытыми кодом бесплатны, имеют объемное сообщество и полную документацию. К их недостаткам можно отнести достаточно долгий промежуток между обновлениями или исправлениями багов, их многокомпонентную структуру, медленное развертывание, недоступность выполнения некоторых функций мониторинга без дополнительных действий или навыков, высокое потребление ресурсов и работа с агентами. Платные решения скомпилированы в одну программу или набор готовых к использованию модулей, способны к быстрому развертыванию и использованию, содержат полный набор необходимых функций. Среди недостатков подобных систем выделяются система ежемесячной/ежегодной подписок, возможность работать лишь с GUI и зависимость от внешней службы поддержки продукта.

Не всем ресурсам локальной сети требуется мониторинг. Его необходимость возникает из поставленных задач, требующих лаконичного подхода. В сетях класса С при лимите до 1000 рабочих

машин количество критически важных компонентов сети крайне мало и не требует использования всех предлагаемых сторонними продуктами функций. Однако такие системы также должны быть способны выполнять базовый набор задач, которые использует в своей работе системный администратор. К ним можно отнести обнаружение устройств в сети, проектирование отчетов об аппаратных комплектующих сетевых устройств и др.

В условиях ограниченности ресурсов кластера, недоступности некоторых функций мониторинга и времени развертывания сторонних продуктов появляется необходимость в создании дополнительных инструментов мониторинга, позволяющих автоматизировать определенные задачи системного администратора. Разработка некоторых из них и методы их внедрения будут рассмотрены на примере локальной сети предприятия СОУНБ им. В.Г. Белинского.

Предлагаемая методика. Создание подобных инструментов в компьютерах под управлением Windows стало возможным благодаря средству автоматизации Powershell от Microsoft. Оно позволяет инкапсулировать данные о системе в объекты и использовать их в дальнейшем при обработке запросов. Powershell – полноценный язык программирования, позволяющий использовать классы других языков программирования и специальные командлеты для доступа к ресурсам операционной системы.

Командлеты – это улучшенный тип функций, которые позволяют определенным образом передавать им аттрибуты и передавать их вывод другим функциям с помощью конвеера. Конвеер Powershell – это уникальная особенность Powershell, которая делает возможным передачу вывода одной команды на вход следующей. Она позволяет выстраивать интересные конструкции и разбивать сложные задачи на несколько этапов.

Средство автоматизации Powershell входит в состав компонентов Windows, однако требует несколько подготовительных этапов перед развертыванием. Среди всех языков программирования количество постов на StackOverFlow занимает лидирующее место в разделе безопасности [3, 4]. Прежде всего необходимо помнить, что политика использования сценариев Powershell в Windows по умолчанию запрещена – Restricted. Для того чтобы выполнить её развертывание, необходимо использовать административные шаблоны ActiveDirectory, либо воспользоваться следующим командлетом: Set-Execution Policy – Execution Policy Remote Signed (Unrestricted).

Рекомендуется оставлять политику Remote Signed, позволяющую использовать написанные вами лично сценарии без запроса сертификатов подтверждения. Политика выполнения скриптов Unrestricted позволяет использовать любые файлы сценариев Powershell, что является опасной практикой.

Для удаленного доступа к другим компьютерам с помощью технологии сеансового уровня, реализованного в Powershell, необходимо добавить на локальном компьютере имена всех удаленных компьютеров, к которым будет выполняться подключение, в список доверенных хостов с помощью определенного командлета с атрибутом:

Set-Item WSMan:\localhost\Client\TrustedHosts – Value (hostname).

Помимо этого, на каждом из удаленных компьютеров должна быть запущена служба WMIили IIS для обработки подобных подключений.

Кроме того, необходимо открыть порт брендмауэра 5985 или выполнить в командной строке команду “winrmquickconfig”. Она создает исключение фаервола для текущего пользователя.

Еще одним шагом при подготовке является синхронизация установленных версий Powershell. В комплекте Windows 10, начиная с 19 версии, уже встроен Powershell 5.1, в то время как на Windows 7 может стоять лишь Powershell 2.0. Реализация технологии WMI для извлечения информации о компьютере появилась лишь начиная с третьей версии, как и некоторые способы вызова циклов. В версиях 1.0–5.1 Powershell считается привязанным к ОС Windows, однако шестая версия Powershell Core является уже кроссплатформенной. Язык Powershell основан на NetFramework и использует классы языка C#. С выходом NetCoreC# стал кросс-платформенным, что перестало привязывать Powershell только к одной операционной системе. Начиная с Powershell 6.0, он стал доступен на MacOs, Windows, Linux [5–7].

После проделанных действий можно писать собственные сценарии под конкретно поставленный набор задач системного администратора. Поскольку они будут выполняться многократно и автоматически, их можно реализовать в виде функций, вызываемых по требованию или необходимости. Ранее уже было сказано, что в языке программирования Powershell есть два типа функций – обычные и улучшенные. Начиная с версии 3.0 языка Powershell, обычные функции перестали использоваться, так как их функционал уже содержится в улучшенных функциях, включающих в себя возможности командлетов.

Для определения необходимых функций нужно сформировать концепцию мониторинга сети и соотнести с техническим заданием предприятия.

Основные этапы сетевого мониторинга представлены на рисунке 1, они будут циклически выполняться для каждого устройства.

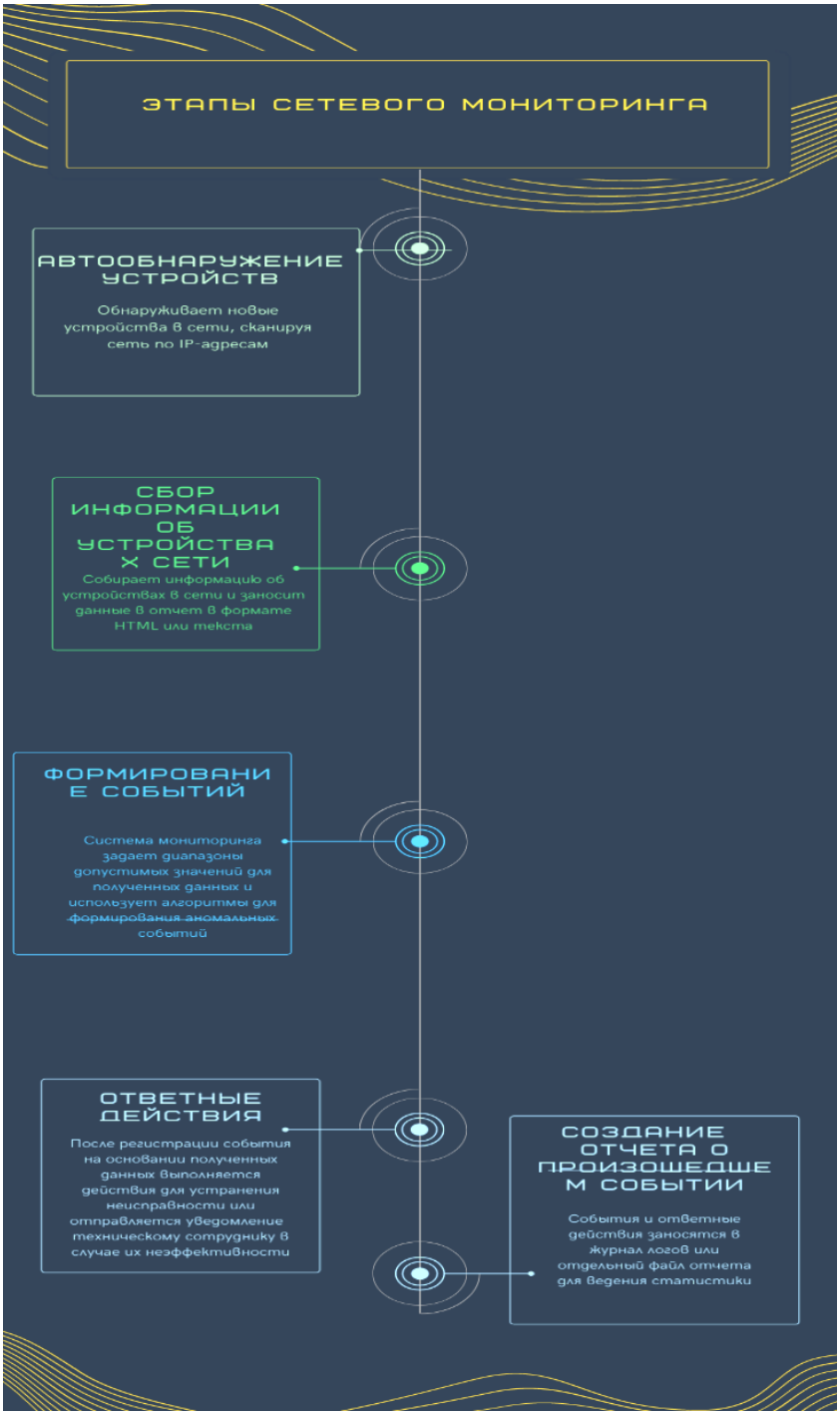


Рисунок 1 – Основные этапы сетевого мониторинга

После осуществления развертывания Powershell и выполнения предварительных действий – установки политики запуска скриптов и заведения списка доверенных хостов – администратор пишет код функций, ориентированный на выполнение конкретных задач на разных этапах:

1. Обнаружение устройств и их экспорт в файл таблицы хранения паролей.
2. Создание отчетов о найденных устройствах в формате HTML.
3. Формирование событий об активных соединениях на TCP-портах удаленного доступа – SSH и RDP.

4. Формирование событий об изменении скорости текущего соединения с внешними доменами.
5. Отправка уведомлений о произошедших событиях в групповой чат Telegram.
6. Добавление записи о произошедшем событии в текстовый файл логов.

7.1. **Функция обнаружения устройств и их экспорт в файл таблицы хранения паролей.** В библиотеке реализованы пять сетей класса С – читательская, для сотрудников, серверная, видеонаблюдения и IP-телефонии. Сеть расположена в домене, но DHCP-роль не установлена. Все данные о пользователях и их паролях хранятся на сетевом ресурсе в таблице. На настоящий момент в библиотеке происходит реорганизация нескольких отделов. Возникла необходимость в автоматизированном обнаружении сетевых устройств и экспорта полученных хостов в табличный файл, в котором будет также храниться информация о пользователях и паролях. Пароли переносятся вручную в целях безопасности, а остальные данные – IP, DNS-имя, MAC, информация о процессоре, имя используемого пользователя – запрашиваются с каждого хоста сети с помощью технологии WMI в цикле for. Диапазон переменной счетчика i задается атрибутами функции – начало и конец диапазона сканирования. Технология WMI представляет собой репозиторий, место в котором разделено на различные адресные пространства для каждого класса устройств. Обратившись к пространству класса, с удаленного компьютера можно получить доступ к атрибутам конкретного устройства. Например, для класса win32_physicalmemory можно запросить любые атрибуты оперативной памяти – частота, объем, используемый слот на материнской плате компьютера. В качестве табличного файла можно использовать Excel или CSV. Последний был выбран из-за поддержки его встроенным модулем Powershell.

7.2. **Функция создания отчетов о найденных устройствах в формате HTML.** Следующая функция запрашивает уже более детальный отчет, используя ту же технологию, но экспортируя заранее отобранные атрибуты во внешний файл формата HTML. До настоящего момента отчеты об аппаратных компонентах компьютера снимались программами Everest и AIDA64 в момент физического нахождения рядом с устройством, чаще всего это происходило в момент диагностики устройства. Они хранились локально и запрашивались при необходимости для отслеживания замены компонентов компьютеров. Эти программы поддерживали и HTML-формат отчетов, но содержали в себе достаточно много лишней информации и имели объем до нескольких десятков мегабайт.

Формат гиперссылок считается более привычным для человеческого глаза и может быть использован на веб-сервере IIS для предоставления статистики. Процесс осуществляется за счет командлета ConvertTo-HTML. В строку вывода этой команды можно добавить html-теги, разбив текст на отдельные блоки. К ним можно применить стили CSS, которые могут быть включены непосредственно в скрипт Powershell или прикрепляться с помощью внешнего файла с расширением CSS. Кроме того, такие отчеты не содержали информацию о производительности устройств в реальном времени – работе их процессов и служб. Сотрудниками отдела технического обеспечения СОУНБ им. В.Г. Белинского был выведен ряд параметров, которые требовалось включить в содержание отчета о рабочих станциях. Заключительный вариант включал в себя три раздела – информация о ключевых аппаратных компонентах, производительность на уровне служб и процессов на момент снятия отчета, а также подключенные к компьютеру принтеры.

Отчеты снимались по пять минут, и информацию из них было крайне трудно извлечь. Это представлено на рисунке 2.

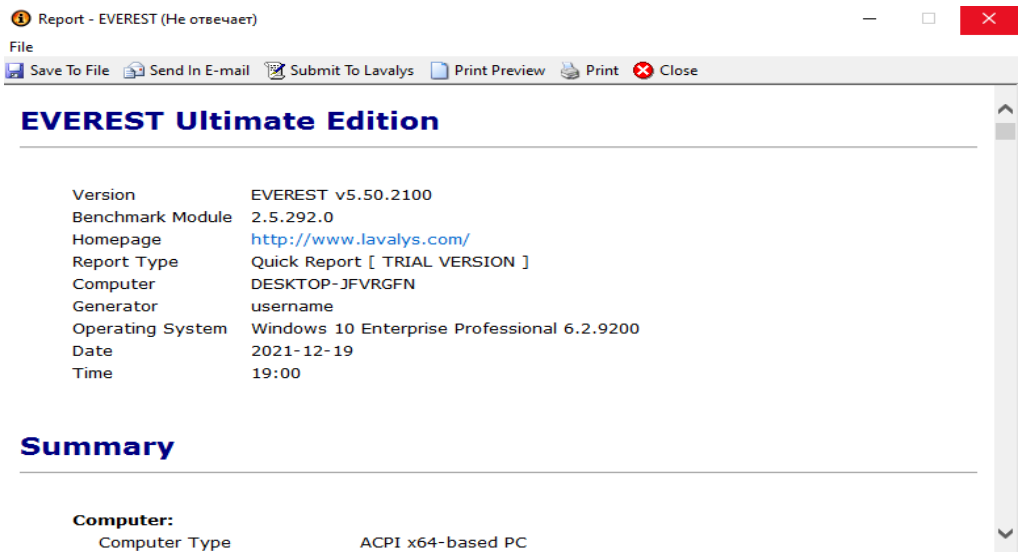


Рисунок 2 – Скриншот отчета информации

7.3. **Функция формирования событий об активных соединениях на TCP-портах удаленного доступа – SSH и RDP**. Эти две функции включают в себя первые два этапа описанной концепции. В качестве формирования событий могут быть использованы параметры статуса работы служб, состояние активных соединений на определенных TCP портах, скорости соединения с внешним доменом. Концепция этой функции построена на открытии удаленной сессии с другим компьютером, его периодическим опросом в бесконечном цикле через определенные интервалы времени.

7.4. **Функция формирования событий об изменении скорости текущего соединения с внешними доменами.** В библиотеке подключены два провайдера – Erlang и Planeta. Скорость тарифов на каждом из них – 30 и 100 Мбит/соответственно. При осуществлении перехода между ними будет сформировано событие, при котором скорость соединения сервера достигнет определенного значения. Для реализации этой функции можно использовать классы C# – Webclient или System.Net.Http.HttpClient из библиотеки SystemNet.

7.5. **Функция отправки уведомлений о произошедших событиях в групповой чат Telegram.** Для быстрого реагирования системному администратору необходима функция уведомлений, которая будет отправлять сообщения не на почту, а на почтовый мессенджер Telegram. Общение между техническими специалистами библиотеки осуществляется с помощью группового чата, потребуется создать бота в телеграме через специальный чат, далее использовать его токен – уникальный идентификатор, набор вызовов API для принятия и отправки сообщений.

7.6. **Функция добавления записи о произошедшем событии в текстовый файл логов.** Необходима функция последнего этапа – запись произошедшего события в логи для составления статистики. Она выполняется с помощью командлета Set-Content с атрибутом добавления строки в файл.

Описание структуры модуля. Пользовательский модуль Powershell – это файл с расширением psm1, в котором расположен листинг всех включенных в него функций. Он хранится в одном из трех стандартных расположений на локальном компьютере. Начиная с третьей версии Powershell достаточно использовать функцию модуля, чтобы все его функции из физического расположения загрузились в сеанс консоли автоматически.

После создания файла модуля можно дополнительно к нему создать файл манифеста модуля с раширением psd1. В нем хранится дополнительная информация о созданном модуле – поддерживаемая версия Powershell, информация об авторе и краткое описание его работы.

Реализация работы первых этапов модуля. После развертывания системному администратору нужно войти в систему под учетной записью с соответствующими правами; сравнить установленную версию Powershell с версией, указанной в манифесте модуля; открыть консоль Powershell или среду ISE, запустив с правами администратора; один раз установить модуль с помощью командлета Install-Module и подтвердить действие в консоли; вызывать и использовать функции этого модуля при составлении других сценариев. При повторном использовании Powershell на этом компьютере устанавливать модуль уже не требуется – достаточно вызвать любую из его функций. Скриншоты работы первых двух функций приведены на рисунках 3–6.

Итоговым этапом будет регистрация модуля в общедоступном репозитории PowershellGallery. Это необходимо для того, чтобы иметь возможность устанавливать его на разных компьютерах с помощью одного командлета Install-Module.

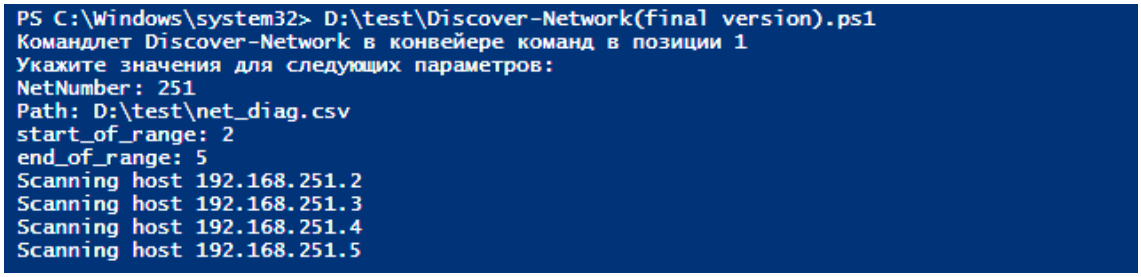
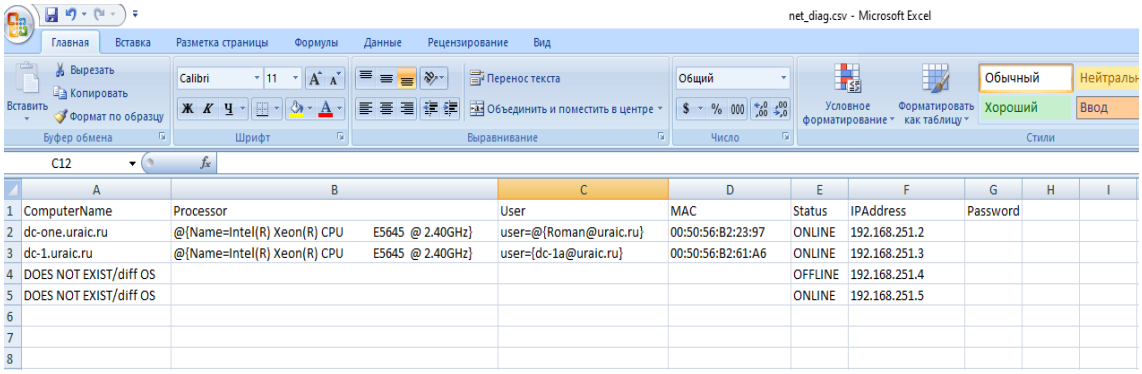


Рисунок 3 – Скриншот выполнения функции обнаружения устройств и их экспорт в файл таблицы хранения паролей



net_diag.csv - Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	ComputerName	Processor	User	MAC	Status	IPAddress	Password		
2	dc-one.uraic.ru	@{Name=Intel(R) Xeon(R) CPU E5645 @ 2.40GHz}	user=@{Roman@uraic.ru}	00:50:56:B2:23:97	ONLINE	192.168.251.2			
3	dc-1.uraic.ru	@{Name=Intel(R) Xeon(R) CPU E5645 @ 2.40GHz}	user={dc-1a@uraic.ru}	00:50:56:B2:61:A6	ONLINE	192.168.251.3			
4	DOES NOT EXIST/diff OS				OFFLINE	192.168.251.4			
5	DOES NOT EXIST/diff OS				ONLINE	192.168.251.5			
6									
7									
8									

Рисунок 4 – Скриншот результата функции обнаружения устройств и их экспорт в файл таблицы хранения паролей

```
PS C:\WINDOWS\system32> hostname
DESKTOP-JFVRGFN

PS C:\WINDOWS\system32> D:\Учеба\ВКР(Powershell)\ИТОГОВОЕ\CreateHTMLreport-Computer.ps1
cmdlet CreateHTMLreport-Computer at command pipeline position 1
Supply values for the following parameters:
ComputerName: DESKTOP-JFVRGFN
Path: D:\Powershell
Making report for DESKTOP-JFVRGFN ...

PS C:\WINDOWS\system32>
```

Рисунок 5 – Скриншот выполнения функции снятия отчета компьютера в формате HTML

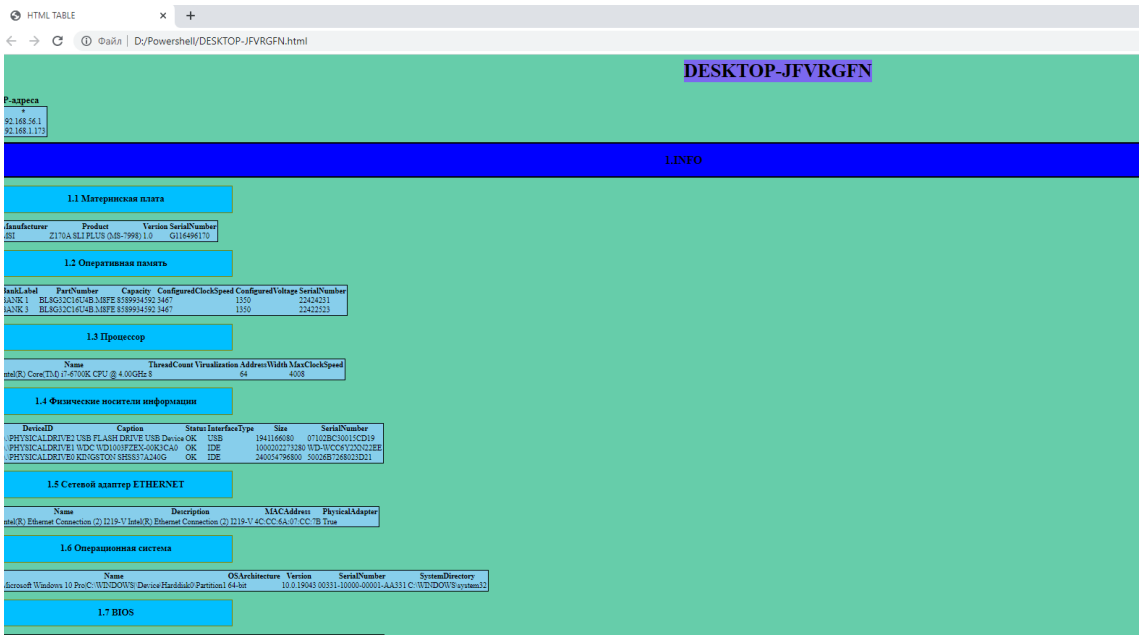


Рисунок 6 – Скриншот результата функции снятия отчета компьютера в формате HTML

Таким образом, этот модуль станет набором программных инструментов для системных администраторов с открытым исходным кодом и дополнит функциональные возможности средства автоматизации Powershell.

Заключение. В работе предложены этапы разработки пользовательского модуля Powershell для мониторинга сети класса С. Проанализированы текущие проблемы в локальной сети библиотеки им. В.Г. Белинского, а также предварительные действия для развертывания Powershell.

Решения задач разработки некоторых функций модуля Powershell и рассмотрения принципа проектирования других осложняются тестированием функции обнаружения устройств в сети и формирования отчета о рабочих станциях под управлением Windows в формате HTML.

Помимо этого, рассмотрен репозиторий пользовательских модулей PowershellGallery. Показана структура работы модуля и порядок действий при его использовании.

Библиографический список

1. Lukas, Macura, Miroslav, Voznak. Multi-criteria analysis and prediction of network incidents using monitoring system / Lukas Macura, Miroslav Voznak // Journal of Advanced Engineering and Computation. – 2019, October. – Vol. 1, № 1. – P. 31–34.
2. Emanuel, Zgârdea, Ladislau, Augustinov. Improving IT Infrastructure Management Using Nagios Open Source Package / Emanuel Zgârdea, Ladislau Augustinov // Analele Universităţii "Eftimie Murgu" Reşiţa: Fascicola I, Inginerie. – 2014, December. – P. 334–336.
3. Sudhakar, Sushil, Kumar. An emerging threat Fileless malware: a survey and research challenges / Sudhakar, Sushil Kumar // Cybersecurity. – 2020, February. – P. 1–4.
4. Croft, Roland. An empirical study of developers' discussions about security challenges of different programming languages / Croft Roland, Xie Yongzheng, Zahedi Mansooreh, Babar M. Ali, Treude // Empirical Software Engineering. – 2021, December. – P. 23–27.
5. The Power Shell Gallery. – Режимдоступа: https://docs.microsoft.com/en-us/powershell/scripting/gallery /overview?view=powershell-7.2, свободный. – Заглавие с экрана. – Яз. англ. (дата обращения: 22.12.2021).
6. Бертрам, А. Powershell для сисадминов / А. Бертрам. – Санкт-Петербург : Издательский дом «Питер», 2021. – 416 с.
7. Олифер, В. Компьютерные сети. Принципы, технологии, протоколы / В. Олифер, Н. Олифер. – 5-е изд. – Санкт-Петербург : Издательский дом «Питер», 2016. – 992 с.

References

1. Lukas, Macura, Miroslav, Voznak. Multi-Criteria Analysis and Prediction of Network Incidents Using Monitoring System. *Journal of Advanced Engineering and Computation*, 2019, October, vol. 1, no. 1, pp. 31–34.
2. Emanuel, Zgârdea, Ladislau, Augustinov. Improving IT Infrastructure Management Using Nagios Open Source Package. *Analele Universităţii "Eftimie Murgu" Reşiţa: Fascicola I, Inginerie*, 2014, December, pp. 334–336.
3. Sudhakar, Sushil, Kumar. An emerging threat Fileless malware: a survey and research challenges. *Cybersecurity*, 2020, February, pp. 1–4.
4. Croft, Roland, Xie, Yongzheng, Zahedi, Mansooreh, Babar, M. Ali, Treude, Christoph. An empirical study of developers' discussions about security challenges of different programming languages. *Empirical Software Engineering*, 2021, December, pp. 23–27.
5. The Powershell Gallery. Available at: https://docs.microsoft.com/en-us/powershell/scripting/gallery /overview?view=powershell-7.2 (accessed 21.12.2021).
6. Bertram, A. *Powershell dlya sisadminov* [Powershell for sysadmins]. Saint-Petersburg, Publishing House "Piter", 2021. 416 p.
7. Olifer, V., Olifer, N. *Kompyuternye seti. Printsipy tekhnologii, protokoly* [Computer networks. Principles, technologies and protocols]. Saint-Petersburg, Publishing House "Piter", 2016. 992 p.

DOI 10.54398/2074-1707_2022_1_113
УДК 004.001

МЕТОД ЗАЩИТЫ СИСТЕМЫ МАШИННОГО ОБУЧЕНИЯ ОТ ВРЕДОНОСНЫХ ПРОГРАММ

Статья поступила в редакцию 01.02.2022, в окончательном варианте – 15.02.2022.

Петренко Вячеслав Иванович, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,

кандидат технических наук, и. о. директора Института цифрового развития, заведующий кафедрой организации и технологии защиты информации, ORCID: 0000-0003-4293-7013, e-mail: vipetrenko@ncfu.ru

Тебуева Фариза Биляловна, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,

доктор физико-математических наук, заведующая кафедрой компьютерной безопасности, ORCID: 0000-0002-7373-4692, e-mail: ftebueva@ncfu.ru

Анзоров Артур Русланович**,** Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,

студент, ORCID: 0000-0002-1157-4021, e-mail: artanzrv@gmail.com

Стручков Игорь Владиславович, Северо-Кавказский федеральный университет, 355017, Российская Федерация, г. Ставрополь, ул. Пушкина, 1,

аспирант, ORCID: 0000-0001-8744-498X, e-mail: selentar@bk.ru

Статья посвящена проблеме защиты системы машинного обучения от вредоносного ПО. Проведен анализ возможных уязвимостей систем машинного обучения, приведена классификация наиболее опасных атак с описанием классов, включающих в себя способ воздействия и последствия от применения данных атак в системе машинного обучения. Для противодействия ряду атак предложен метод защиты системы машинного обучения от вредоносных программ на основе алгоритмов Neural-Cleanse и Jpeg-Compression.

Ключевые слова: машинное обучение, нейронные сети, информационная безопасность, Neural-Cleanse, Jpeg-Compression, атаки отравления, атаки искажения, атаки извлечения модели