ҚАЗАҚСТАН РЕСПУБЛИКАСЫ ҒЫЛЫМ ЖӘНЕ ЖОҒАРЫ БІЛІМ МИНИСТРЛІГІ

«Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ» КЕАҚ

Студенттер мен жас ғалымдардың
«ǴYLYM JÁNE BILIM - 2025»
XIX Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ
XX Международной научной конференции
студентов и молодых ученых
«ǴYLYM JÁNE BILIM - 2025»

PROCEEDINGS
of the XX International Scientific Conference
for students and young scholars
«ǴYLYM JÁNE BILIM - 2025»

2025
Астана

УДК 001(06)
ББК 72я631
F96

«ǴYLYM JÁNE BILIM – 2025» студенттер мен жас ғалымдардың XX Халықаралық ғылыми конференциясы = XX Международная научная конференция студентов и молодых ученых «ǴYLYM JÁNE BILIM – 2025» = The XX International Scientific Conference for students and young scholars «ǴYLYM JÁNE BILIM – 2025». – Астана: – 3813 б. - қазақша, орысша, ағылшынша.

ISBN 978-601-08-5373-7

Жинаққа студенттердің, магистранттардың, докоранттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.В сборник вошли доклады студентов, магистрантов, докторантов имолодыхученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 001(06)
ББК 72я431
F96

ISBN 978-601-08-5373-7

Л.Н. Гумилев атындағы Еуразия ұлттық университеті, 2025

СЕКЦИЯ 2

СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ И КОММ УНИКАЦИОННЫЕ ТЕХНОЛОГИИ

Подсекция 2.2

Интеллектуальные информационные системы

255.	Сафонова Софья Александровна «Современные аспекты информационной безопасности в облачных вычислениях: модели, угрозы и методы защиты»	1034
256.	Смаилова Назгүл Батырбекқызы «Терең оқыту арқылы кітап ұсыныстарын әзірлеу: collaborative filtering, content-based және nlp әдістерінің комбинациясы»	1041
257.	Тажібай Аружан Айдосқызы, Кудубаева Сауле Альжановна «Көру қабілеті әлсіз адамдарға арналған аі дауыстық көмекші: нақты уақытта объектілерді анықтау және қашықтықты бағалау»	1046
258.	Тайжанов Азамат Жанкелдіұлы «Python тілінде фильмдердің интеллектуалды ұсыныс жүйесін әзірлеу»	1051
259.	Умирзахов Сундетали Кабылбекович «Сұраныстарды интеллектуалды талдау негізінде ұйымның сайты үшін чат-бот құру»	1055
260.	Шайхстан Марғұлан «IOT Сенсорлары негізінде ауа ластану деңгейін болжау»	1060

Подсекция 2.3

Современные тенденции в программной инженерии и управлении в условиях цифровой индустрии

261.	Bekenova A.B. «Development of a registration panel for users and doctors with integration into the database»	1077
262.	Bolat A.Zh. «Data analysis methods and decision making using big data and machine learning tools»	1081
263.	Алтайұлы А. «Visual studio интегралды ортасында «қойма қызметкерлеріне арналған» мәліметтер қорын жобалау»	1086
264.	Арап А.Қ. «Ақылды сурет салушы роботты әзірлеу»	1088
265.	Артыкбекқызы А. «Ақылды үйлердегі заттар интернеті(iot) мен робототехниканың өзара әрекеттесуі»	1091
266.	Ахметова А.Д. «Тоңазытқыштағы өнімдерді бақылауға және тағам әзірлеу ұсынысын беруге арналған программалық қосымша»	1096
267.	Дәрібай Д.Д. «Робототехниканы қолдану арқылы қойма логистикасындағы қолданыстағы басқару жүйелерін талдау»	1100
268.	Жамбулов С.Ж. «Білім алушыларды информатика және программалау олимпиадаларына дайындауда жасанды интеллекттің қолданысы»	1102
269.	Каиржан Р.С. «Development of system for recognition of emotional states of employees based on computer vision methods on Raspberry Pi»	1108
270.	Кайрекенова Н.Р. «Өнеркәсіптік роботты көру үшін машиналық оқытудың заманауи тәсілдері: әдістер, деректер жиынтығы және оптимизациялау»	1111

СЕКЦИЯ 3
ЕСТЕСТВЕННЫЕ НАУКИ

- TLS/SSL пайдалану: барлық WebSocket қосылымдары деректердің құпиялылығы мен тұтастығын қамтамасыз ету үшін шифрланған арналарды (wss://) пайдалану;
- Сайтаралық сценарийден қорғау (XSS): пайдаланушы енгізуін санитариялау, Content Security Policy (CSP) қауіпсіздік тақырыптарын орнату және басқа шаралар арқылы XSS-тің алдын алу;
- CSRF шабуылдарынан қорғау: веб-пішіндер мен басқа күйді өзгертетін сұраулар үшін CSRF шабуылдарына қарсы белгілерді пайдалану;
- Аутентификация және авторизация: OAuth, JWT немесе HTTP-Only және Secure cookie сеанстарын пайдалануды қоса алғанда, барлық кезеңдерде сенімді аутентификация мен авторизацияны қамтамасыз ету;

Пайдаланылған дереккөздердің тізімі

1. WebSocket: Lightweight Client-Server Communications, Andrew Lombardi 2015, 141p
2. The Definitive Guide to HTML5 WebSocket, Frank Salim Peter Moskovits, 2013, 227 p
3. https://cyberleninka.ru/article/n/kak-ispravit-mezhsaytovuyu-uyazvimost-web-socket
4. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9182458
5. https://book.hacktricks.xyz/pentesting-web/websocket-attacks
6. https://medium.com/swlh/hacking-websocket-25d3cba6a4b9
7. https://portswigger.net/web-security/websockets
8. https://ably.com/topic/websocket-security
9. https://www.shodan.io/
10. https://datatracker.ietf.org/doc
11. https://infosecwriteups.com/cross-site-websocket-hijacking-cswsh-ce2a6b0747fc/html/rfc6455
12. https://infosecwriteups.com/cross-site-websocket-hijacking-cswsh-ce2a6b0747fc

УДК 004.056

АВТОМАТИЗАЦИЯ ПРОЦЕССА АНАЛИЗА ОПЕРАТИВНОЙ ПАМЯТИ С ИСПОЛЬЗОВАНИЕМ PYTHON

Майданов Алихан Серикович
Alikhan.m.01@gmail.com
Магистрант кафедры Информационная безопасность ЕНУ им. Л.Н. Гумилева, Астана, Казахстан Астана, Казахстан
Научный руководитель – Ахметова Ж.Ж.

Аннотация. В данной работе будет рассмотрен **Volatility Framework** и его ключевые возможности, связанные с анализом оперативной памяти. Будут продемонстрированы примеры использования плагинов, связанных с выявлением аномалий в процессах, закрепившихся в оперативной памяти, анализ подгруженных dll-файлов и их указателей, а также исследование оперативной памяти на предмет внедренного кода. Более того, в данной статье рассмотрены правила Yara Rules, а также скрипты на Python, позволяющие автоматизировать процесс анализа памяти. Результатом исследования станет создание набора автоматизации на базе Python, которые позволят автоматизировать рутинные задачи по анализу оперативной памяти, для ускорения процесса расследования кибератак и повышения его точности.

Ключевые слова: кибербезопасность, Volatility Framework, цифровая криминалистика, анализ оперативной памяти, автоматизация анализа памяти, аномалии в процессах, внедренный код, Yara Rules.

Введение

Современные киберугрозы становятся все более сложными и изощренными, что делает цифровую криминалистику важным направлением в области информационной безопасности. Одним из ключевых аспектов цифровой форензики является анализ оперативной памяти, поскольку именно в памяти можно обнаружить артефакты вредоносной активности, которые не всегда сохраняются на диске. Исследование дампа оперативной памяти позволяет получить критически важные данные, такие как процессы, сетевые соединения, зарегистрированные учетные записи, загруженные библиотеки и даже зашифрованные коммуникации вредоносных программ.

В этом контексте Volatility Framework является одним из наиболее популярных и высокофункциональных инструментов для анализа памяти. Volatility Framework — это платформа с открытым исходным кодом, предназначенная для анализа оперативной памяти различных операционных систем, включая Windows, Linux, macOS и Android. Она предоставляет обширный набор плагинов, позволяющих исследовать процессы, драйверы, файлы, сетевые соединения, реестр операционных систем, а также извлекать вредоносный код из памяти. Возможности Volatility делают его незаменимым инструментом в расследованиях кибератак, анализе вредоносного программного обеспечения (ПО) и реагировании на инциденты безопасности.

Однако, несмотря на функциональность Volatility, процесс исследования оперативной памяти остается достаточно сложным и требует значительных временных затрат. Запуск различных плагинов, анализ их результатов и поиск подозрительной активности вручную — трудоемкая задача. Кроме того, анализ может включать обработку больших объемов данных, что делает его подверженным человеческим ошибкам.

Анализ оперативной памяти является одной из ключевых задач цифровой криминалистики и расследования инцидентов информационной безопасности. Однако традиционные методы анализа требуют значительных временных и вычислительных ресурсов, а также высокой квалификации специалиста. Дамп памяти содержит огромный объем информации: активные процессы, загруженные библиотеки, сетевые соединения, открытые файлы, системные артефакты и даже следы вредоносной активности. Обнаружение атакующих действий в этом массиве данных вручную — сложная, трудоемкая и подверженная ошибкам задача.

Материалы и методы

В этом контексте автоматизация анализа памяти с помощью Python-скриптов и Volatility Framework становится необходимостью. Она позволяет сократить время расследования, исключить человеческий фактор при анализе и выявлять аномалии быстрее и точнее.

Volatility Framework – это мощный инструмент цифровой криминалистики, предназначенный для анализа дампов оперативной памяти (RAM). Он позволяет исследовать активные процессы, сетевые соединения, загруженные библиотеки (DLL), следы вредоносного ПО и многое другое [1].

Volatility был разработан как проект с открытым исходным кодом и распространяется под лицензией GNU General Public License (GPL). Это означает, что любой может использовать, изменять и распространять код. Разработчики могут создавать собственные плагины и расширять функциональность. Пример использования Volatility: python vol.py -f memDump.bin windows.pstree

Плагин `pstree` в Volatility 3 используется для **построения дерева процессов, запущенных в момент создания дампа памяти.** Он помогает определить родительские и

дочерние процессы, что особенно полезно при расследовании вредоносной активности, так как позволяет выявить: а**номальные процессы, которые не соответствуют типичной иерархии, процессы, запущенные без родительского процесса или с необычными родителями, использование утилит для повышения привилегий или выполнения команд удаленно.**

Плагин psxview в Volatility 3 используется для обнаружения скрытых процессов в дампе оперативной памяти. Этот инструмент помогает обнаружить процессы, которые могут быть замаскированы вредоносным ПО (например, руткитами) и не отображаются в стандартных списках процессов Windows.

На рисунках ниже представлены выводы команд дерева процессов (Рисунок 1)

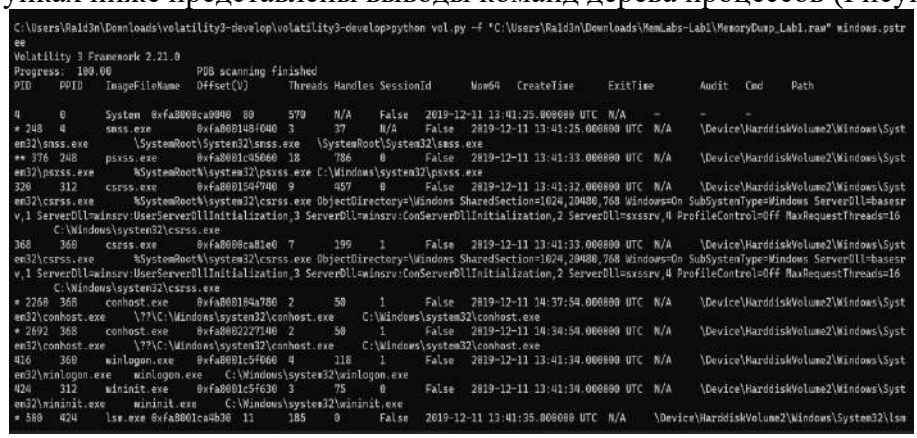

Рисунок 5 Вывод команды pstree

Одним из наиболее эффективных инструментов в Volatility является модуль Malfind, предназначенный для поиска следов внедренного кода в адресном пространстве процессов. Вредоносные программы часто используют различные техники инъекций, позволяя атакующим исполнять произвольный код в доверенных процессах Windows.

Принцип работы модуля Malfind заключается в том, что он анализирует виртуальные адресные пространства процессов и выявляет аномалии, указывающие на возможные инъекции кода. Основные критерии, по которым Malfind обнаруживает подозрительные участки памяти, включают:

1. Выделение памяти с правами исполнения (PAGE_EXECUTE_READWRITE) — вредоносный код часто размещается в таких участках.
2. Отсутствие привязки памяти к загруженным модулям — если область памяти не принадлежит ни одной из известных DLL, это может быть признаком внедрения.
3. Наличие исполняемых инструкций в неожиданном месте — например, если процесс загружает код, не относящийся к его стандартному исполняемому файлу.

Вывод плагина включает в себя: PID процесса, в который была выполнена инъекция, виртуальный адрес подозрительного участка памяти, фрагмент содержимого памяти в шестнадцатеричном представлении (hex dump), дизассемблированный код [2].

Другим важным инструментом анализа оперативной памяти является **Dlllist** — модуль, позволяющий получить список загруженных динамических библиотек (**DLL**) для каждого активного процесса [3].

Загрузка вредоносных **DLL** является одной из распространенных техник маскировки вредоносного ПО. Атакующие могут использовать **DLL Injection** для выполнения своего кода внутри легитимных процессов Windows. Анализ загруженных библиотек помогает выявить такие угрозы, как:

1. **Использование нелегитимных DLL в системных процессах**.
2. **Наличие неизвестных или подозрительных библиотек в процессе**.

3. **Отсутствие подписей у загруженных модулей**.

Применение Volatility в расследованиях помогает не только выявлять вредоносную активность, но и автоматизировать процесс анализа с помощью Python-скриптов, что особенно актуально для работы с большими объемами данных [4].

Результаты

На рисунке ниже приведена основная функция, которая используется в написанном автором коде (Рисунок 3).

```
def run_volatility(plugin, dump_file):
    """Запуск Volatility 3 и вывод соотв. информации."""
    cmd = f"python vol.py -f {dump_file} {plugin}"
    result = subprocess.run(cmd, shell=True, capture_output=True, text=True)
    return result.stdout
```

Рисунок 3 Функция запуска Volatility

На основе команды python vol.py -f будут впоследствии запускатся различного рода плагины для дальнейшего исследования. Эта функция **автоматизирует работу с Volatility 3**, упрощая анализ дампов памяти. Она позволяет запускать плагины, получать их результаты и использовать их в Python-скриптах для автоматизации цифровой криминалистики.

Данный код является открытым исходным кодом. В python-скрипте будут реализованы следующие плагины для автоматизации исследования оперативной памяти:

```
# Плагины Volatility которые будут запущены
plugins = {
    "Processes (pslist)": "windows.pslist",
    "Hidden Processes (psscan)": "windows.psscan",
    "Hidden and stealthy processes": "windows.psxview",
    "Processes in tree format": "windows.pstree",
    "Malicious Injections (malfind)": "windows.malfind",
    "Loaded DLLs (dlllist)": "windows.dlllist",
```

Рисунок 6 Плагины Volatility

Суть автоматизации данного процесса, в том, что он позволяет избежать рутинных действий и помогает извлекать необходимые для исследования цифровые улики в рамках расследования произошедших атак или инцидентов информационной безопасности.

Обсуждение

Скрипт помогает извлечь все необходимые артефакты, в один клик. К тому же, стандартный вывод выполнения данных плагинов отображается в командной строке, что немного неудобно при длительном изучении. Данный скрипт извлекает информацию и записывает ее в .csv и .txt форматы. В случае, исполнения плагинов pslist и dlllist, в которых вывод представляет собой список, данный скрипт сохраняет их в .csv формате для удобства чтения, а вывод остальных команд сохраняется в .txt. Терминальный вывод работы скрипта, изображен ниже (Рисунок 5).

```
Running Volatility Plugins...
 Запуск плагина: Processes (pslist) (windows.pslist)...
 CSV отчет сохранен как Processes_(pslist)_output.csv
--------------------------------------------------------------------------------
 Запуск плагина: Hidden Processes (psscan) (windows.psscan)...
 Вывод плагина сохранен как Hidden_Processes_(psscan)_output.txt
--------------------------------------------------------------------------------
 Запуск плагина: Hidden and stealthy processes (windows.psxview)...
 Вывод плагина сохранен как Hidden_and_stealthy_processes_output.txt
--------------------------------------------------------------------------------
 Запуск плагина: Processes in tree format (windows.pstree)...
 Вывод плагина сохранен как Processes_in_tree_format_output.txt
--------------------------------------------------------------------------------
 Запуск плагина: Malicious Injections (malfind) (windows.malfind)...
 Вывод плагина сохранен как Malicious_Injections_(malfind)_output.txt
--------------------------------------------------------------------------------
 Запуск плагина: Loaded DLLs (dlllist) (windows.dlllist)...
 CSV отчет сохранен как Loaded_DLLs_(dlllist)_output.csv
--------------------------------------------------------------------------------
 Отчет сохранен в файле summary_report.json
```

Рисунок 7 Терминальный вывод скрипта

При выводе работы скрипта, помимо сохранения данных в файлы, с определенным названием плагинов, дополнительно создается файл в .json формате, чтобы удостовериться какие именно файлы были созданы.

Также ниже приведен вывод команд malfind. Далее по ходу расследования, данные файлы могут сыграть важную роль в нахождении подозрительно подгруженных DLL-библиотек и внедренного кода в различные процессы.

```
Volatility 3 Framework 2.21.0

PID     Process Start VPN       End VPN Tag     Protection      CommitCharge    PrivateMemory

668     csrss.exe       0x850000        0xb4ffff        Vad     PAGE_EXECUTE_READ       0
c1 00 00 00 00 01 00 00 ff ee ff ee 09 00 00 00 ................
09 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 ............. ..
00 02 00 00 00 20 00 00 50 33 00 00 ff ef fd 7f ..... ..P3.....
00 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 ................        c1 00 00 00 00 01 00
00 50 33 00 00 ff ef fd 7f 00 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00
668     csrss.exe       0xbd0000        0xc4ffff        Vad     PAGE_EXECUTE_READ       0
c1 00 00 00 00 01 00 00 ff ee ff ee 09 00 00 00 ................
09 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 ............. ..
00 02 00 00 00 20 00 00 b0 0c 00 00 ff ef fd 7f ..... ..........
00 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 ................        c1 00 00 00 00 01 00
00 b0 0c 00 00 ff ef fd 7f 00 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00
668     csrss.exe       0x7f6f0000      0x7f7effff      Vad     PAGE_EXECUTE_READWRITE  0
c8 00 00 00 13 01 00 00 ff ee ff ee 08 70 00 00 .............p..
08 00 00 00 00 fe 00 00 00 00 10 00 00 20 00 00 ............. ..
00 02 00 00 00 20 00 00 8d 01 00 00 ff ef fd 7f ..... ..........
03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00 ................        c8 00 00 00 13 01 00
00 8d 01 00 00 ff ef fd 7f 03 00 08 06 00 00 00 00 00 00 00 00 00 00 00 00
```

Рисунок 8 Файловый вывод плагина malfind

Заключение

Автоматизация процесса анализа дампов оперативной памяти является важным шагом в ускорении и повышении точности цифровых расследований. Использование скриптов на Python для интеграции Volatility позволяет значительно снизить временные затраты на ручной анализ, а также уменьшить вероятность ошибки за счет стандартизации подходов к обработке данных.

В данной работе было проведено исследование возможностей автоматизации анализа оперативной памяти с использованием Python и Volatility Framework. Рассмотрены ключевые плагины, такие как pstree, psxview, malfind и dlllist, которые позволяют детектировать аномальные процессы, скрытые или модифицированные исполняемые файлы, внедренный вредоносный код и загруженные библиотеки DLL.

В ходе исследования были разработаны программные инструменты, позволяющие автоматизировать запуск Volatility и обработку его вывода. Такой подход позволяет эффективно идентифицировать потенциальные угрозы, анализировать активность вредоносного ПО и получать структурированную информацию о состоянии системы на момент создания дампа памяти.

Результаты работы подтверждают, что внедрение автоматизированных решений на основе Python и Volatility Framework может существенно повысить эффективность работы специалистов по цифровой криминалистике.

Список использованной литературы

1. Cohen, M (2020). Incident Response & Computer Forensics, Third Edition. McGraw-Hill Education.
2. Sikorski, M., & Honig, A. (2022). Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. No Starch Press.
3. Ligh, M. H., Case, A., Levy, J., & Walters, A. (2021). The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory. Wiley.
4. Case, A., & Richard, G. (2019). Memory Forensics: The Path Forward. Digital Investigation, 29, S6-S17.

УДК 004.75.

БЛОКЧЕЙН ҚАЖЕТТІЛІК ПЕ, ӘЛДЕ СӘН БЕ?

Мақсат Әсемай, Нурсеитов Сейіл
nurseitovseil@gmail.com, ***maksatasemai001@gmail.com***
Л. Н. Гумилев атындағы ЕҰУ Ақпараттық технологиялар факультетінің
Ақпараттық қауіпсіздік кафедрасының 3 курс студенттері,
Астана, Қазақстан
Ғылыми жетекші – Казиева Назым Магидулловна

Аңдатпа: Мақалада блокчейн қажет пе? деген сұрақты талдадық. Тақырыпты зерттеу барысында басқа модельдерге сүйене отырып, блокчейн қажет пе, егер блокчейн қажет болған жағдайда, қандай түрді қолдану керек екендігі туралы нақты және дұрыс түсінік алу үшін маңызды ойлар мен сұрақтарды бөлдік.

Түйінді сөздер: блокчейн, криптовалюта, тізілім (реестр), тест-сауалнама.

Кіріспе. Мақалада қазіргі әлемдегі блокчейн технологиясының өзектілігі мен қажеттілігі қарастырылады. Орталықтандырылмаған шешімдерге деген қызығушылықтың артуы жағдайында блокчейн қаржы, логистика, денсаулық сақтау және деректерді басқаруды қоса алғанда, әртүрлі салаларға жаңа мүмкіндіктер ұсынады [1]. Жұмыста блокчейннің негізгі принциптері, оның артықшылықтары мен кемшіліктері, сондай-ақ нақты жобаларда сәтті қолдану мысалдары талданады [2]. Блокчейн әртүрлі салаларда қамтамасыз ете алатын қауіпсіздік, ашықтық және тиімділік мәселелеріне ерекше назар аударылады.

Талдау аясында осы технологияны кәсіпорынның барлық бизнес-процестерінде қолдану қажеттілігі бар-жоғын бағалауға, сондай-ақ олардың міндеттерінің ерекшелігіне байланысты қолайлы блокчейн моделін таңдауға мүмкіндік беретін тест әзірледік. Тест әзірлеу барысында әртүрлі модельдерді зерттеп, сол модельдерге сүйендік. Бұл модельдер туралы қысқаша тоқталып өтейік.