УДК 81’23

СИСТЕМЫ KEYLOGGER В ОРГАНИЗАЦИИ ДЛЯ АНАЛИЗА НЕСАНКЦИОНИРОВАННЫХ ДЕЙСТВИЙ

Хачетлов Мухамед Асланбиевич, студент,
Арванова Саният Мухамедовна, старший преподаватель,
Нагоев Мухамед Амирбиевич, студент,
ФГБОУ ВО «КБГУ», г. Нальчик

KEYLOGGER SYSTEMS IN ORGANIZATIONS FOR ANALYZING UNAUTHORIZED ACTIONS

Khachetlov Mukhamed Aslanbievich, student, myxamed077123@gmail.com,
Arvanova Saniyat Mukhamedovna, Senior Lecturer, sani_07@mail.ru,
Nagoev Mukhamed Amirbievich, student, nagoevm01@gmail.com,
Kabardino-Balkarian State University, Nalchik, Russia

Разработана система KeyLogger на Python для мониторинга активности пользователей в корпоративных средах с целью предотвращения утечек данных и несанкционированного доступа. Система фиксирует ввод с клавиатуры, отслеживает приложения, использует шифрование для защиты журналов. Позволяет выявлять подозрительные действия (доступ к запрещенным ресурсам, передача конфиденциальной информации), обеспечивает прозрачность при минимизации избыточного сбора данных. Интегрируется с DLP-системами, перспектива включает внедрение ИИ для автоматического обнаружения аномалий. Балансирует контроль и этические нормы за счет ограниченного доступа к данным.

Ключевые слова: *KeyLogger, информационная безопасность, предотвращение утечек данных, корпоративная среда, мониторинг активности.*

A KeyLogger system has been developed in Python to monitor user activity in corporate environments, aimed at preventing data leaks and unauthorized access. The system records keyboard input, tracks applications, and employs encryption to protect logs. It detects suspicious actions (e.g., accessing restricted resources, transferring confidential data) while ensuring transparency and minimizing redundant data collection. The solution integrates with DLP systems, with future prospects including AI implementation for automated anomaly detection. It balances control and ethical standards by restricting data access.

Keywords: *KeyLogger, information security, data leak prevention, corporate environment, activity monitoring.*

В современных условиях, когда информационные технологии играют ключевую роль в функционировании организаций, вопросы информационной безопасности становятся всё более актуальными. Одной из наиболее серьёзных угроз для корпоративной среды являются несанкционированные действия сотрудников, которые могут привести к утечке конфиденциальной информации, нарушению политик безопасности или даже к финансовым потерям. В связи с этим возникает необходимость в разработке и внедрении инструментов, способных отслеживать и анализировать действия пользователей на рабочих станциях, чтобы своевременно выявлять потенциальные угрозы и предотвращать их [5].

Одним из таких инструментов является система KeyLogger, которая позволяет фиксировать все действия пользователя, включая ввод с клавиатуры, использование приложений и другие активности. KeyLogger может быть использован как для мониторинга производительности сотрудников, так и для выявления подозрительных действий, таких как попытки доступа к запрещённым ресур-

сам, передача конфиденциальных данных или нарушение корпоративных политик. Однако применение таких систем требует тщательного баланса между контролем и соблюдением этических норм, чтобы не нарушать права сотрудников на приватность [1].

Целью данной работы является разработка и внедрение системы KeyLogger в корпоративной среде для анализа несанкционированных действий сотрудников. Основные задачи нашего исследования включают:

1. Разработку программного решения на языке Python, способного фиксировать и анализировать действия пользователей на рабочих станциях;
2. Внедрение механизмов шифрования и ограниченного доступа к данным для обеспечения конфиденциальности собранной информации;
3. Создание системы фильтрации и анализа данных, позволяющей выявлять потенциально опасные действия и минимизировать сбор избыточной информации.

Разработанная система KeyLogger представляет собой инструмент, который может быть интегрирован в корпоративную инфраструктуру для повышения уровня информационной безопасности. Она позволяет администраторам получать полную картину активности пользователей, выявлять подозрительные действия и предотвращать утечки данных. При этом система обеспечивает прозрачность и контроль, что способствует соблюдению корпоративных политик и повышению дисциплины среди сотрудников [2].

В рамках настоящей работы была разработана система KeyLogger, ориентированная на анализ действий пользователей в корпоративных сетях с целью предотвращения несанкционированных операций. Основу системы составляет программное обеспечение, реализованное на языке Python, которое обеспечивает запись ввода с клавиатуры, мониторинг активности и сохранение данных в зашифрованном формате.

Для реализации ключевых функций системы использованы стандартные библиотеки Python: os (работа с файловой системой), time (учёт временных меток), logging (ведение журнала событий), datetime (генерация временных отметок), а также сторонняя библиотека pynput.keyboard для отслеживания нажатий клавиш. Интеграция этих инструментов позволила обеспечить базовую функциональность системы [4].

Система автоматически создаёт директорию для хранения логов, используя метод os.makedirs с параметром exist_ok=True, что исключает ошибки при повторном создании папки. Логирование настроено через модуль logging с уровнем DEBUG для фиксации всех событий. Формат записей включает временные метки, уровень сообщения и его содержание, а данные сохраняются в файл с уникальным именем, сгенерированным на основе текущей даты и времени (например, день-месяц_час-минута.log).

Ядро системы — функция on_press, которая обрабатывает каждое нажатие клавиши. Для буквенно-цифровых символов используется атрибут key.char, для специальных клавиш (например, Ctrl, Space) — key.name. Все события записываются в лог-файл через log_file.write и параллельно фиксируются в журнале с помощью logger.debug.

Система функционирует в бесконечном цикле с проверкой периода бездействия пользователя. При отсутствии активности в течение 30 секунд текущий лог-файл закрывается, что позволяет разграничивать сеансы работы. Для снижения нагрузки на процессор добавлена задержка time.sleep(0.1).

Запуск системы осуществляется через командный файл start_keylogger.cmd, который автоматически активирует Python-скрипт в фоновом режиме. Это позволяет программе стартовать без видимого интерфейса, что особенно удобно для интеграции в автозагрузку операционной системы или для задач, требующих минимального вмешательства в рабочий процесс. После инициализации кейлоггер переходит в режим мониторинга, отслеживая каждое нажатие клавиш и сохраняя данные в дебаг-файле (Рис.1) и лог-файле (Рис. 2) [3].

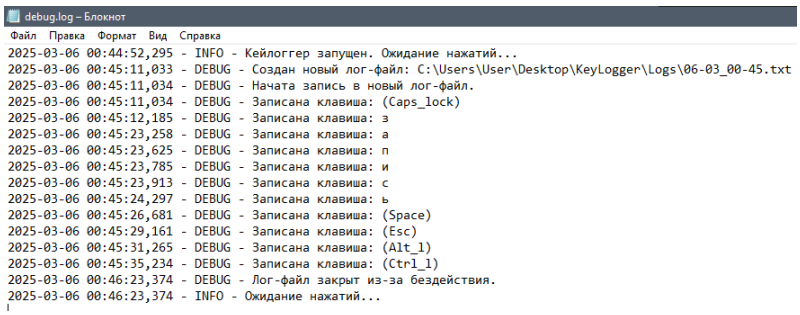

Рисунок 1. – Отлатка процессов в файл debug.log

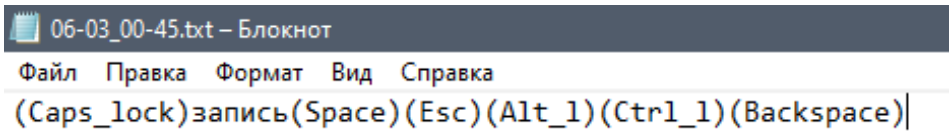

Рисунок 2. – Запись нажатий в файл с расширением .txt

Сразу после запуска в файле debug.log появляется запись:

ГГГГ-ММ-ДД ЧЧ:ММ:СС - INFO - Кейлоггер запущен. Ожидание нажатий...

Это подтверждает корректную инициализацию системы. Если пользователь не проявляет активность в течение 30 секунд, программа автоматически закрывает текущий лог-файл, фиксируя это событием:

DEBUG - Лог-файл закрыт из-за бездействия.

При возобновлении активности создаётся новый файл с уникальным именем, содержащим временную метку (например, 15-07_14-30.log), что упрощает анализ сессий и поиск аномалий.

Система фиксирует не только буквенно-цифровые символы, но и специальные клавиши. Например, нажатие пробела регистрируется как (Space), а активация Caps Lock — как (Caps_lock). Это позволяет воссоздавать полную картину действий пользователя, включая использование служебных клавиш, что критично для анализа подозрительной активности.

Текущая версия обеспечивает базовый функционал, однако её можно усовершенствовать. Например, сохранение логов в скрытую директорию через os.hidden повысит незаметность системы. Для централизованного сбора данных с нескольких устройств можно настроить путь к сетевой папке, а интеграция библиотеки smtplib позволит автоматически отправлять логи на почту, упрощая удалённый мониторинг.

Разработанная система KeyLogger подтверждает возможность эффективного использования Python для создания инструментов мониторинга, адаптированных под корпоративные требования. Её архитектура сочетает автономность работы и гибкость: запуск через командный файл обеспечивает фоновое функционирование без вмешательства в пользовательский опыт, а генерация лог-файлов с уникальными временными метками позволяет систематизировать данные для последующего анализа.

Особое внимание уделено детализации записей: система фиксирует не только стандартные символы, но и специальные клавиши, такие как Space или Caps_lock, что критично для точного воспроизведения действий пользователя. Автоматическое закрытие логов при 30-секундном бездействии оптимизирует хранение данных, снижая нагрузку на ресурсы.

Этические аспекты внедрения требуют строгого контроля. Использование кейлоггера допустимо только при условии прозрачности: уведомление пользователей о мониторинге и соблюдение законодательных норм становятся обязательными элементами внедрения.

Перспективы развития системы связаны с повышением её универсальности. Например, сохранение логов в скрытые директории или сетевые папки расширит возможности

Список использованных источников

1. Ермаков А. В. Компьютерная криминалистика: методика расследования компьютерных преступлений. - М.: Инфра-М, 2009. – 368 с

2. Бурмистров А. П. Компьютерная криминалистика. Учебное пособие. – М.: Горячая линия – Телеком, 2014. – 368 с

3. Смирнов А. П. Python для кибербезопасности. – М.: Диалектика, 2017. – 320 с

4. Петрухин В. А. Практическая цифровая криминалистика: методы и инструменты расследования инцидентов информационной безопасности. – СПб.: Питер, 2015. – 400 с

5. Козлов И. Н. Кибербезопасность в корпоративной среде: теория и практика. – М.: Бином, 2018. – 352 с