DOI: 10.58168/PBHT2026_213-216
УДК 004.056.53

МОДЕЛИ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ EBPF ДЛЯ СКРЫТНОГО НАБЛЮДЕНИЯ ЗА СИСТЕМОЙ В LINUX

MODELS OF UNAUTHORISED USE OF EBPF FOR SECRET SYSTEM MONITORING IN LINUX

Синявский А.П., курсант 3 курса, Федеральное казенное образовательное учреждение высшего образования «Воронежский институт Федеральной службы исполнения наказаний России», Воронеж, Россия

Sinyavsky A.P., 3rd-year cadet, Federal Treasury Educational Institution of Higher Education "Voronezh Institute of the Federal Penitentiary Service of Russia", Voronezh, Russia

Лубенцов А.В., кандидат географических наук, доцент, профессор кафедры основ радиотехники и электроники, Федеральное казенное образовательное учреждение высшего образования «Воронежский институт Федеральной службы исполнения наказаний России», Воронеж, Россия

Lubentsov A.V., PhD in Geographical Sciences, Docent, Professor of the Radio Engineering and Electronics Fundamentals Department, Federal Treasury Educational Institution of Higher Education "Voronezh Institute of the Federal Penitentiary Service of Russia", Voronezh, Russia

Аннотация. В статье рассматривается технология eBPF в ОС Linux, позволяющая безопасно выполнять программы внутри ядра. Анализируется двойственная природа данного механизма: несмотря на статический анализ и обеспечение безопасности самого выполнения кода, злоумышленники активно используют его для создания скрытных руткитов и вредоносного ПО. Отмечается, что подобные угрозы работают на уровне ядра без создания процессов в системе, перехватывают системные вызовы и практически не обнаруживаются традиционными средствами защиты.

Ключевые слова: ОС Linux, eBPF, Ядро системы, Безопасность, Руткиты, Системные вызовы.

Abstract. The article discusses the eBPF technology in the Linux operating system, which allows for the secure execution of programs within the kernel. It analyzes the dual nature of this mechanism: despite the static analysis and security of the code execution itself, attackers actively use it to create stealthy rootkits and malware. It is noted that such threats operate at the kernel level without creating processes in the system, intercepting system calls and being virtually undetectable by traditional security measures.

Keywords: Linux operating system, eBPF, System kernel, Security, Rootkits, System calls.

В современные операционные системы (ОС) требуют все более высокоэффективные механизмы мониторинга, анализа и фильтрации событий, вводиться сложные и эффективные алгоритмы. Так в ОС Linux одним из наиболее актуальных инструментов такого рода явиться eBPF (extended Berkeley Packet Filter).

© Синявский А. П., Лубенцов А. В., 2026

eBPF – это механизм ядра Linux, позволяющий загрузить и выполнять программы внутри ядра. Как указывает официальная документация вся деятельность данного совета основана на системных триггерах bpf, которые и обеспечивают операции над eBPF-программами и картами, при этом сам код строго проверяется ядром. То есть ядро проводит статический анализ программы перед их загрузкой, чтобы гарантировать безопасных исход дальнейших манипуляций с ядром. В частности, верификатор eBPF обеспечивает конечность выполнения, что ведет к исключению появления бесконечных циклов, а также он предоставляет возможность полного исключения доступов и ограничений стека. Благодаря этому eBPF-программы безопасно выполняются вплоть до заверения и не вызывают так называемой паники ядра [1].

Однако такая гибкость и надежность делает данную систему так называемым двуликим объектом в плане безопасности. С одной стороны, хотя в данный момент используется множество современных инструментов для защиты, с другой же стороны злоумышленники могут использовать eBPF для скрытого контроля и фильтрации данных в ядре системы. Как. Отмечено в исследовании AquaSec, за последний год появилась тенденция к злоупотреблению eBPF, а именно с помощью данной системы активно создают ракиты такие как TripleCross, и вредоносные программы, такие как Pamspy Которые похищают учётные данные и обходят средства защиты. При этом традиционны средства обнаружения часто бессильны против данных угроз из-за сложно доступности самого кода, протекающего через ядро, который считается неприкосновенным или наиболее защищенным. Из-за этого создаться так называемая иллюзия чистоты, то есть файловый след после таких атак практически невозможно обнаружить, что делает неволимыми и сами руткиты, и их действия [2].

Вся это системы работает на основе получения привилегий даваемых eBPF-программ для взаимодействия с ядром, что позволяет глубоко вмешиваться в работы самой системы Linux. В отличии от других руткитов, руткиты построены на основе eBPF работают в самом ядре, то есть он не записывается в кэш и ему вообще не принадлежит какой-либо процесс в системе, что дает буквально: «контроль в самом корне». Это позволяет, например, имплементировать в ядре скрытный брандмауэр или фильтр трафика без следов в пользовательском пространстве.

Для большего понимания модно сказать, что данная система чем-то похожа по своей работе с BIOS руткитами. То есть eBPF руткиты могут подменять системные процессы и файлы буквально до их передачи самой программы [3].

Кроме сетевого уровня eBPF позволяет перехватывать и модифицировать системные вызовы. Чаще всего злоумышленники подменяют вывод системы инструментов. Например, eBPF-программа может прихватывать системный вызов getdents64 (используется при ls и ps) и удалить из возвращаемого списка файлы или процессы, связанные с руткитом.

То есть в данном случае системы имеет фундаментальную уязвимость, которую практически невозможно обнаружить, поскольку eBPF-программы живут в ядре. Традиционны Linux-дистрибутивы ориентированы на файловую систему и пользовательский процессы, они не сканируют память ядра и не видят загруженные eBPF-процессы. Это приводит к тому, что без защиты на уровне ядра нельзя доверять ничему, даже выводам инструментов диагностирования.

Данная проблема несет критический уровень опасности, так как система, зараженная данными руткитами, может считаться практически мертвой. То есть защитой системы необходимо заниматься еще до потенциального попадания данного руткита в ядро. Следует настроить аудит и мониторинг системных вызова и использовать дополнительные модули безопасности, основанные на аудите системного вызова bpf, подавляющее большинство злоумышленных eBPF-программ, которые как раз и импортируются в ядро через bpf. Аналогично этому стоит так же отслеживать ubshare, Clone3 или другие используете функции [4].

Еще одним шагом к построению защиты от таких рэкетов служит ограничение доступа самой bpf, конечно это тонкая грань функционала и безопасности, но она несомненна выглядит как необходимая мера. Так же установка сессионной проверки целостности ядра служит хорошим вариантом защиты, поскольку стандартные внтивирусы малоэффективны против eBPF. Рекомендуется использовать модули обеспечения целостности, такие как LKRG (Linux Kernel Runtime Guard). С помощью него мы сожжёт полноценно дать понимать системе все взаимодействия с ядром.

В некоторых источниках так же предлагается для полной полноты картины обеспечить систему цифровыми подписями и проверкой kexec. То есть в этом случае собственно программы eBPF перед запуском проверяться, чтобы ядро принимало только так называемых доверенный код [5].

В совокупности эти меры создают многоуровневую защиту системы с аудирование, ограничением прав и привилегий, проверкой целостности и активным мониторинга, что и даст уверенность о защите глубинного модуля Linux. При использовании всех рекомендованных методов и механизмов риск незаметного скрытого мониторинга eBPF значительно снижается. Однако важно помнить, что ответственность за безопасность системы не может быть полностью возложена на одно средство защиты. Любое послабление в виде вовремя не обновлённого ядра могут давать те самые уязвимости, которые могут привести утечки персональных и секретных данных с служебных персональных компьютеров.

СПИСОК ЛИТЕРАТУРЫ

1. Евдокимов, А.В. Архитектура и защита ядра операционных систем семейства Linux: учеб. пособие / А.В. Евдокимов. – Москва : ИНФРА-М, 2022. – 315 с.

2. Макаров, В.В. Использование технологии eBPF для создания и обнаружения руткитов в современных ОС / В.В. Макаров, Д.А. Смирнов // Проблемы информационной безопасности. – 2022. – № 4. – С. 55-62.

3. Сидоров, П.Н. Анализ верификатора eBPF и возможные векторы атак на ядро Linux / П.Н. Сидоров // Инновации в области защиты информации : материалы Всероссийской научно-практической конференции. – Санкт-Петербург, 2023. – С. 112-118.

4. Иванов, И.И. Обзор современных вредоносных программ на базе eBPF: Pamspy и TripleCross / И. И. Иванов // Вестник кибербезопасности. – 2023. – № 2 (10). – URL: http://cybervestnik.ru/2023/02/pdf/14.pdf (дата обращения: 10.04.2026).

5. Руководство по архитектуре BPF (Berkeley Packet Filter) : официальная документация ядра Linux. – URL: https://www.kernel.org/doc/html/latest/bpf/index.html (дата обращения: 10.04.2026).

6. Detecting and mitigating eBPF-based threats // Aqua Security : [офиц. сайт]. – 2023. – URL: https://www.aquasec.com/research/ebpf-threats (дата обращения: 10.04.2026).

7. Марков, А.С. Перехват системных вызовов в Linux: эволюция от модулей ядра (LKM) до eBPF / А. С. Марков // Информационные технологии и телекоммуникации. – 2021. – Т. 9, № 3. – С. 89-97.

REFERENCES

1. Evdokimov, A.V. Architecture and Protection of the Kernel of Linux Operating Systems: a tutorial / A.V. Evdokimov. – Moscow: INFRA-M, 2022. – 315 p.

2. Makarov, V.V. Using eBPF Technology to Create and Detect Rootkits in Modern OS / V.V. Makarov, D.A. Smirnov // Problems of Information Security. – 2022. – No. 4. – P. 55-62.

3. Sidorov, P.N. Analysis of the eBPF Verifier and Possible Attack Vectors on the Linux Kernel / P.N. Sidorov // Innovations in Information Security: Proceedings of the All-Russian Scientific and Practical Conference. – St. Petersburg, 2023. – P. 112-118.

4. Ivanov, I.I. Review of Modern eBPF-Based Malware: Pamspy and TripleCross / I.I. Ivanov // Cybersecurity Bulletin. – 2023. – No. 2 (10). – URL: http://cybervestnik.ru/2023/02/pdf/14.pdf (accessed: 10.04.2026).

5. BPF (Berkeley Packet Filter) Architecture Guide: official Linux kernel documentation. – URL: https://www.kernel.org/doc/html/latest/bpf/index.html (accessed: 10.04.2026).

6. Detecting and mitigating eBPF-based threats // Aqua Security: [official website]. – 2023. – URL: https://www.aquasec.com/research/ebpf-threats (date of access: 10.04.2026).

7. Markov, A.S. Interception of system calls in Linux: evolution from kernel modules (LKM) to eBPF / A.S. Markov // Information Technologies and Telecommunications. – 2021. – Vol. 9, No. 3. – P. 89-97.