ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ МОНИТОРИНГА, УКРЕПЛЕНИЯ И ТЕСТИРОВАНИЕ БЕЗОПАСНОСТИ GNU LINUX

Антон СЕНЮШИН

Департамент компьютерных наук и системной инженерии, CRI-231M, факультет вычислительной техники, информатики и микроэлектроники, Технический университет Молдовы, Кишинев, Республика Молдова

Автор кореспондент: Антон СЕНЮШИН, e-mail: seniusin.anton@iis.utm.md

Научный руководитель/координатор **MORARU Victor**, conf. univ

Резюме. *Данная научная статья посвящена практическому исследованию мониторинга, укрепления и тестирования защиты системы Linux, которая содержит в себе теоретическое и практическое объяснение структуры системы. В свою очередь структура описывает создание программного обеспечения рассчитанного на мониторинг файловой системы и мониторинг параметров сервера для помощи неопытным администраторам и пользователем на начальном пути администрирования серверов. Так вся система связанна с Linux, то все необходимые зависимости устанавливаются на, по мере прохождения проекта, дистрибутив Debian. Первая часть проекта реализована в виде frontend, которая отвечает за связь и активацию между компонентами всего проекта, реализованная на web framework Flask. Вторая часть проекта заключается в действиях происходящих на backend, которая в свою очередь отвечает за два массивных модуля: мониторинг файловой системы и мониторинг параметров сервера в режиме реального времени. Теоретическая часть отвечает за объяснение важности понимания кибербезопасности и необходимых тактик защиты, подкреплённых практическими навыками. Весть проект создан по принципам современной защиты серверов, а также укрепления защиты при помощи программного обеспечения с открытым кодом.*

Ключевые слова: *linux, cybersecurity, python, docker, flask, virtualization*

Введение

Один из основных концептов, который приобрёл особую популярность в последние годы заключается во внедрении безопасности, а именно укрепление и последующий мониторинг производимых проектов. С резким ростом использования технологий, также увеличилось количество атак проводимых на запущенных в производство проектах. Следовательно необходимость в профессионалах кибер безопасности как никогда требуется на рынке труда, что следственно ведёт к более ужесточённым условиям по вхождению в данную область. Для решения данной проблемы существует множество систем и технологий, которые позволяют упростить укрепление и мониторинг системы даже не имея при этом богатого опыта работы с системами Linux и Windows [1].

Система предлагаемая в данной статье разработана при помощи математической модели и реализована на базе операционной системы Linux (имитируя серверную среду) используя технологию виртуализации docker, с графическим интерфейсом для упрощения укрепления и мониторинга системы начинающим пользователем или/и администраторам. Так как одной из больших проблем с которой сталкиваются начинающие администраторы это опознание стороны, с которой произойдёт атака, следовательно фактор виртуализации помогает правильно понять текущую проблему и вернуть состояние системы до изначального положения.

Так безопасность системы довольно тяжело описать, по причине существования множества факторов, которые влияют на производительность, способность выдержать

высокую нагрузку, распределение привилегий и остальные внешние и внутренние переменные. Следовательно разработанная теоретическая и практическая модель приблизительно включает в себя множество основополагающих факторов безопасности системы. Как ранее упоминалось, кроме фактора виртуализации в данную систему входит изобилие вероятностей и коэффициентов, которые определяют общий коэффициент безопасности системы и затем используется в расчёте увеличенного значения общей сохранности архитектуры от возможной проводимой атаки. Следственно укрепление и мониторинг безопасности GNU Linux, используя графический интерфейс, упрощаются, а также в случае атаки будет сохранено текущее состояние контейнеров для дальнейшего анализа.

Включая множество факторов безопасности и удобный для работы интерфейс, разработанная система позволит легче начать понимать такой глобальный вопрос как кибербезопасность и укрепление совместно с мониторингом системы GNU Linux.

Этапы исследования или как правильно понять кибербезопасность

Для проведения практического исследования мониторинга, тестирования, а также укрепления системы GNU Linux необходимо для начала произвести анализ существующих технологий, которые также или частично относятся к базовым понятиям самой системы Linux, так и произведения базовых операций по улучшения базовой защиты и мониторинга, что в свою очередь ссылает на исследование концепта безопасности в сфере операционных система на базе распределения прав доступа, проверка пакет проходящих через сеть, проверка аунтефицированных пользователей и изобилие сторонних и внутренних факторов. Следовательно в конечном итоге, после получения сведений о существующих продуктах, а также концептах безопасности, возможно использовать разработанное ПО (программное обеспечение) в целях безопасного тестирования с последующим восстановлением данных или защиты системы от наружнего вредоносного вмешательства. Также важно использовать именно системы с открытым кодом, что позволит лучше понять и начать адаптироваться под текущую среду GNU Linux.

При проведении анализа актуальности данной темы, можно получить высокие показатели числа кибератак проведённых по предыдущим годам, а также дальнейший прогноз увеличения различных типов атак по экспоненциально-линейной функции Рис 1., что соответственно доказывает большую роль укрепления и мониторинга установленной системы [2].

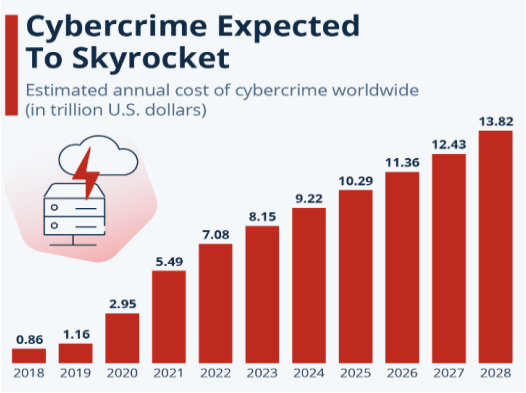

Рисунок 1. Статистика кибератак по всему миру с дальнейшей прогрессией

Такое растущее число кибератак связано с развитием облачных технологий и IOT, а также широкое распространения в инфраструктуре. Следовательно такие вид как DDOS,

MITM и Phishing встречаются всё больше и больше, в конечном итоге причиняя колоссальный вред, как в потере репутации, так и денежном аналоге.

Последовательным решением будет внедрение таких систем как Packet Sniffing Tool, хорошим вариантом в данном случае будет SolarWinds [3], а также Wireshark [4] как бесплатный аналог, далее систему предотвращения и обнаружения вторжений с открытым исходным кодом для анализа трафика в реальном времени и регистрацию файлов Snort, и более простым аналогом Fail2Ban [5]. Но большим минусом данных технологий является отдельная настройка и ограниченное использование в определённом функционале, что может вызвать сложность использования у новичков и начинающих администраторов. Большое преимущество архитектуры предлагаемой в данной статье это большой спектр функционала и простое управление при помощи графического интерфейса, с дополнительным технологиями для укрепления системы при помощи SELinux [6], а также мониторинга системой Prometheus [7] с графическим отображением при помощи Grafana [8]. Также большим преимуществом является изоляция от основной системы при помощи контейнеризации Docker, что добавляет дополнительный слой защиты поверх основной системы.

Использование инструментов и систем является главным шагом к пониманию безопасности системы GNU Linux, но также важно понимать, что существуют основные концепты укрепления системы безопасности, которые включают в себя 3 основных пункта. Первый принцип «Наименьших привилегий» указывает на правильное разделение системы на пользователей, а именно под определённый процесс изменяется право чтения, записи или исполнения, так как каждый процесс, системная запись или учётная запись получают доступ только к той части, в которой необходимо проводить дальнейшие действия. Второй принцип «Сегментация» используется при правильном распределении и применении памяти, а именно каждому процессу выделяется столько памяти сколько необходимо для реализации функционала, так как во время выполнения операций может произойти переполнение памяти или Buffer Overflow, что приведёт к неправильной адресации с последующим получением доступа к информации не предназначенной для данного процесса. Третий принцип «Сокращение» подразумевает удаление всех не использующихся или устарелых библиотек, программ и утилит, что помогает уменьшить вероятность появления уязвимости системы или проникновения с обходом систем безопасности. Применяя все три концепта на практике, возможно с большой вероятностью избежать дальнейших проблем связанных с уязвимостями и неправильным распределением памяти или системных записей [9]. Также в один из основных концептов также может входить «Постоянное обновление», что подразумевает обновление используемых программ или библиотек на новые версии/патчи, так как в более старых версиях могли обнаружиться уязвимости или эксплойты, которые в конечном итоге могут быть использованы для вымогательства денежных средств (Ransomware) или полного удаления данных.

Как начинающий исследователь кибербезопасности обязательно понимать, что данные концепты не просто как рекомендации, а необходимость для внедрения в систему. И после проведения исследования по укреплению, мониторингу и тестированию системы Linux, можно разработать собственную архитектуру на базы проектов с открытым исходным кодом, которая будет включать в себя вышеперечисленные концепты. Система которая описывается в данной статье имеет теоретическое закрепление, так и реализован концепт на практике, который показывает функциональное исполнение и применение виртуализации при помощи контейнеров и методов защиты от наружных и внутренних факторов во время тестирования.

Математическая модель архитектуры как способ доказательства существования

Применяя ранее используемые концепты и общее описание системы укрепления, мониторинга и тестирования возможно получить архитектуру механизма усиления безопасности. Данная архитектура содержит в себе многие возможные вероятности, как отрицательного, так и положительного характера. Каждая вероятность исходит от общих технологий и концептов, которые могут применяться в каждой структуре кибербезопасности. Также важным элементом является коэффициент, отвечающий важности вероятности в системе, так как конечные цели могут отличатся от системы к системе, следовательно, возможно применить базовые понятия логики Fuzzy, чтобы определить важность той или иной вероятности для каждого конкретного случая. Что в следствие может помочь в разработке и использовании нейронной сети совместно с антивирусной утилитой для получения общего балла состояния защиты текущей системы.

Как доказательство эффективности данной системы возможно вывести математическую модель, которая включает в себя большинство вероятностей происхождения тех или иных происшествий и действий. Коэффициент определения безопасности S, будет выглядеть следующим образом Рис. 2.:

S = K1 * P auth - K2 * P access + K3 * P detect + K4 * P firewall - K5 * P vulnerabilities + K6 * P incident_response + K7 * P account_management + K8 * P compliance

Рисунок 2. Коэффициент определения безопасности

Для получения коэффициента определения безопасности были включены многие вероятности происхождения тех или иных действий связанных с системой. Каждый коэффициент Kn получается при помощи логики Fuzzy [10], определяя важность каждой вероятности для конкретной системы, при этом значение коэффициента может варьироваться от 0.1 до 0.95 определяя приблизительное определения данного параметра в используемой системе. При этом данные обозначения расшифровываются следующим образом:

- P auth - вероятностью успешной аутентификации пользователя при попытке доступа к системе, следовательно Ec. (1):

$$P_{auth} = \frac{N_{auth}}{N_{total}} \quad (1)$$

где N auth - количество успешных попыток аутентификации, а N total - общее количество попыток аутентификации.

- P access - вероятность успешного доступа к ресурсу.
- P detect - вероятностью обнаружения атаки
- P firewall - вероятность успешной блокировки сетевых атак
- P vulnerabilities - вероятность успешной эксплуатации уязвимостей.
- P incident_response - эффективность реакции на безопасностные инциденты.
- P account_management - эффективность управления учетными данными.
- P compliance - степень соответствия стандартам безопасности.

Следовательно в конечном итоге получаем коэффициент содержащий сумму параметров относительно всех сторон безопасности, которые могут быть инициированы в каждой системе. Полученный коэффициент определяет безопасность системы GNU Linux, но чтобы добиться максимального значения необходим опыт работы и более глубокие познания самой операционный системы. Решением данной проблемы является использование математической модели проектируемой системы укрепления, мониторинга и тестирования. Так все вероятности входят в множество коэффициента S, то следственно разработанная система автоматизирует данные процессы и добавляет фактор виртуализации с встроенными политиками безопасности Ec (2).

$$C = S + K * SL + V_{container}(W + DB) + V_{container}(M) \quad (2)$$

В конечном итоге практически реализованный концепт C, является суммой ранее представленных вероятностей с коэффициентами важности в связи с вероятностью срабатывания политики безопасности SL и коэффициентов срабатывания K (равный примерно 0.90), также добавление суммы вероятности уязвимости используемой технологии W (пример Apache [11]) и вероятности уязвимости технологии базы данных DB (пример MySQL[12]) под фактором виртуализации с помощью технологии контейнеров Vcontainer, с добавлением вероятности обнаружении пользователем аномалии M, также в виртуализованном пространсте.

Proof of Concept и «с чем его едят»

Кроме внедрения теоретической части, самым лучшим доказательством представляет себя практическая часть. Практическая часть включает в себя создание системы мониторинга для операционной системы, последующее внедрение модуля изменения политик функционирования с организацией архитектуры взаимодействия системы и как последний шаг тестирование. Весь концепт реализован на основе дистрибутива Debian с ядром Linux, используя технологию реверсивного прокси NGINX, систему виртуализации Docker [13], изменение политики безопасности Selinux, и язык программирования Python с библиотеками gunicorn и Flask. Последующее тестирование проведено в ином дистрибутиве в той же подсети KaliLinux [14] Рис. 3.

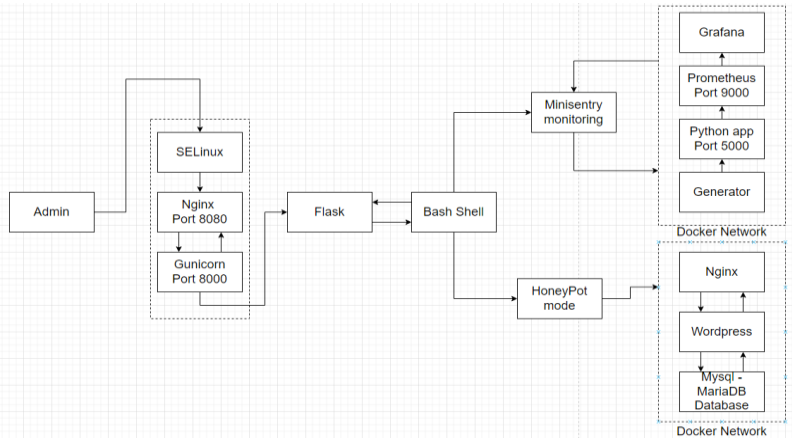

Рисунок 3. Архитектура системы MiniSentry

Работа концепта заключается виртуализации пространства работы технологий установленных на сервер. Тестирование проходило на технологии WordPress с базой данных MySQL, так как без должностной настройки данные технологии уязвимы для внешних атак. Следственно на первых рядах настроена система политик безопасности, что уже ограничивает доступ к системе не авторизованного пользователя. Далее в случае обхода firewall и политик безопасности атакующий попадает в изолированное пространство и начинает изменение встроенных технологий. Тем временем активированная система мониторинга и режима «HoneyPot» фиксирует изменения в системе ранее не зарегистрированные, что в свою очередь подаёт сигнал на активацию защитного протокола. Происходит сохранение текущего состояния и затем блокируется IP адрес с которого была произведена атака. После система восстанавливает данные на основе заранее созданного backup, что приводит в действие механизм сохранение, выключение и повторного включения контейнеров. Заражённый или изменённый контейнер сохраняется для дальнейшего расследования.

Выводы

Представляя данную модель для практического использования новичкам в кибербезопасности и начинающим администраторам, позволяет создать псевдо-лабораторию для экспериментов и автоматической защиты данных уязвимых технологий. Упрощая дальнейшее получения опыта в данной области с последующим поиском дополнительной информации. Включая множество аспектов укрепления, мониторинга и тестирования в одну структуру.

Благодарность.

При компановке данной статьи было потрачено довольно много времени на создание теоретического концепта, поэтому выражаю свою благодарность Морару Виктору, Карбуне Виорелу и Ротару Лилии за мотивацию и поддержку в написании данного материала.

Библиография.

[1] V. Moraru, S, Moraru “Securitatea informaţională ca o condiţie a libertăţii” Revista de Filosofie, Sociologie şi Ştiinţe Politice Numărul 1(164) / 2014 / ISSN 1957-2294

[2] Infographic: Cybercrime Expected To Skyrocket in Coming Years». *Statista Daily Data* [Online]. Available: https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027.

[3] *Observability and IT Management Platform | SolarWinds*. [Online]. Available: https://www.solarwinds.com/.

[4] «Wireshark Go Deep». *Wireshark* [Online]. Available: http://localhost:3000/.

[5] *Fail2ban | Русскоязычная документация по Ubuntu*. [Online]. Available: https://help.ubuntu.ru/wiki/fail2ban.

[6] «SELinux – описание и особенности работы с системой. Часть 1». *Хабр* [Online]. Available: https://habr.com/ru/companies/kingservers/articles/209644/.

[7] Prometheus. *Overview | Prometheus* [Online]. Available: https://prometheus.io/docs/introduction/overview/.

[8] «Grafana | Query, Visualize, Alerting Observability Platform». *Grafana Labs*, [Online]. Available: https://grafana.com/grafana/.

[9] *CentOS 5 Administration – 43.2. Introduction to SELinux*. [Online]. Available: https://www.linuxtopia.org/online_books/centos5/centos5_administration_guide/centos5_ch-selinux.html.

[10] «Нечеткая логика — математические основы». *Loginom.ru* [Online]. Available: https://loginom.ru/blog/fuzzy-logic.

[11] *Documentation Project - The Apache HTTP Server Project*. [Online]. Available: https://httpd.apache.org/docs-project/.

[12] Getting Started». *Oracle Help Center* [Online]. Available: https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started.html.

[13] «Home». *Docker Documentation* [Online]. Available: https://docs.docker.com/.

[14] «Kali Docs | Kali Linux Documentation». *Kali Linux* [Online]. Available: https://www.kali.org/docs/.