РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ДОКЛАД

на тему «Использование Honeypot для защиты от Brute Force атак»

Дисциплина: Администрирование сетевых подсистем

Студент: Генералов Даниил

Группа: НПИбд-01-21

МОСКВА

2022 г.

Содержание

1 Введение

В интернете много плохих людей. Любой компьютер, который публично доступен в интернете, подвергается кибератакам каждый день, и почти половина всех организаций сообщают об атаках против них как минимум раз в месяц [3]. В IPv4-диапазоне всего 4 миллиарда адресов, и с высокой скоростью интернета такое количество легко просканировать всего за 45 минут [6]. Любой компьютер с открытыми портами может представлять интерес: будь то веб-сервер с админской панелью, FTP-сервер с бэкапами базы данных клиентов, сам сервер базы данных, или просто RTSP-сервер у камеры видеонаблюдения на предприятии – можно найти много чего интересного, если просто поискать.

Самые ценные службы в интернете – это SSH-сервера [5]. Они предоставляют пользователям доступ к консоли компьютера. Администраторы используют это, чтобы настраивать сервера, и при этом не ходить к физическому компьютеру. Иногда это единственный способ получить доступ к серверу, потому что его не существует физически, и он на самом деле сделан из виртуальной машины. Если получить доступ к консоли сервера, то можно сделать что угодно: снести сайт, украсть файлы, сделать частью ботнета. Поэтому это исключительно важная цель для хакеров: на моем домашнем сервере я отслеживаю в среднем 5300 попыток взлома за день.

В интернете действительно много плохих людей. К счастью, большинство плохих людей – не люди. Все эти попытки – от ботов, которые просто пробуют частые комбинации логинов-паролей. Они попробуют один-два раза, и если у них не получается то они уходят. Это – так называемая “фоновая радиация” интернета [22]: из-за того, что адресов в IPv4 немного и они постоянно сканируются, любой ботнет легко может найти ваш сервер, и поэтому он должен быть защищен хотя бы от такого уровня попыток взлома. Для SSH один из самых простых способов это сделать – отключить авторизацию по паролю, и разрешать использование только входа по публичному ключу. Это невозможно подобрать, а также фальшивый

сервер не сможет украсть ваш ключ, в отличии от пароля. Все эти попытки взломать мой сервер не смогут преуспеть, потому что мой сервер требует моего ключа.

Но иногда становится интересно: а если бы они смогли зайти в систему, что бы они делали тогда? Этим вопросом задаются исследователи кибербезопасности, чтобы отслеживать тренды в кибератаках, анализировать вирусы, которые используются при этом, и знать как выявлять и предотвращать такие атаки в важных системах. Для таких целей создаются намеренно уязвимые сервера, которые тщательно отслеживаю Хакеры увидят их и клюнут на них, словно медведи на чашу меда, и не заметят ловушки. Именно поэтому такие системы называются “honeypot”, чаша меда [10].

2 Типы Honeypot

Honeypot-продукты существуют для каждого типа сервера, от баз данных до админских веб-панелей. Мы будем обсуждать только SSH-honeypot, потому что это одна из самых частых атак, хотя другие типы honeypot также довольно распространены.

Есть несколько аспектов их использования. Во-первых, если я исследователь кибербезопасности, то полезно отслеживать потенциальные атаки, что можно безопасно делать с помощью изолированной системы, специально предназначенной для того чтобы быть взломанной. Во-вторых, если у нас есть корпоративная сеть, в которой есть также настоящие сервера, и мы не хотим, чтобы их взломали, то мы можем эти сервера защитить правильно, а также поставить honeypot, который будет отвлекать внимание хакеров от по-настоящему важных систем. Наконец, точно так же как с телефонным спамом, есть задача задержать хакеров – чем больше времени они говорят с тобой, тем меньше времени они говорят с другими, кому они могут навредить сильнее.

2.1 Классификация по интерактивности

Ввиду этих разных целей, есть несколько классов honeypot-продуктов, которые различаются по уровню интерактивности. В самом простом случае, мы можем просто подключать входящие соединения к чему-то, что только выглядит как консоль, но на самом деле не обрабатывает команды, или просто всегда выдает ошибку, или бесконечно ждет; или никогда не принимать соединения, но записывать все логины и пароли которые кто-то пробует использовать. Это так называемые низко-интерактивные honeypot. Примеры таких продуктов – `ssh-auth-logger` [9], `ssh-honeypotd` [19].

Слегка сложнее – системы, которые симулируют консоль с базовыми утилитами и виртуальной файловой системой, но которые не связаны с настоящим компьютером. В частности, такие системы могут симулировать, будто хакер смог скачать вирусную программу, но внутри них нельзя будет ее запустить. Это полезно, потому что это может обмануть больше ботов: если бот видит, что консоль не появляется или на каждую команду дает ошибку, то он сразу отключится, а если он сможет выполнить какие-то команды, то он может задержаться. Это называют honeypot средней интерактивности, потому что они могут вести себя слегка правдоподобно, но их можно отличить от настоящего компьютера. Примеры таких продуктов – Kippo [4], Cowrie [13], `sshsyrup` [14].

Наконец, honeypot высокой интерактивности – с точки зрения соединения неотличимы от обычного компьютера. Именно они полезнее всего для исследовательских задач, потому что они дают ботам возможность полностью выполнить свою атаку, и исследователи затем могут разобраться, из чего она состоит. Способов достичь такого результата несколько – например, можно просто дать доступ к какой-то виртуальной (или даже физической) машине, которая находится на изолированном сегменте сети. Можно также использовать продукты вроде `wetland` [15], `LazySSH` [20], `ContainerSSH` [8], которые по запросу создают новую виртуальную машину или контейнер, внутри которого можно делать что угодно.

3 Defence-in-Depth

Honeypot также могут использоваться как компонент более широкой системы безопасности. Например, есть продукты вроде `fail2ban` [18], которые отслеживают логи ошибок авторизации и блокируют IP-адреса, которые там встречаются слишком часто.

Есть распределенная версия этого подхода, называется CrowdSec [2] – здесь, если один и тот же IP-адрес встречается в логах нескольких машин одновременно, то он автоматически блокируется у всех машин, которые используют этот продукт. Тот же самый подход можно использовать с honeypot; более того, можно сказать, что каждая из машин такой сети служит как honeypot для всех остальных. Поэтому такие решения часто рекомендуются для организаций малого масштаба (вроде домашних серверов и других self-hosted служб), которые также не могут вкладывать время на отслеживание киберугроз.

Как и с другими аспектами кибербезопасности, следует рассматривать риски, которые могут привносить honeypot. Использование виртуальных машин, в которых хакеры могут делать что угодно, является самым рискованным подходом, и поэтому это используют только организации, которые профессионально занимаются анализом кибератак. Для более тщательной изоляции, они также часто покупают отдельные, никак не связанные друг с другом VPS, но это требует значительных инфраструктурных затрат, поэтому обычные организации не делают такого.

4 Риски использования

Подход, основанный на виртуальных машинах, дает наиболее точные результаты для анализа, но он также наиболее опасный, потому что если хакер сможет найти способ обойти механизм изоляции (а существуют атаки и на файрволы [17], и на контейнеры [11], и на виртуальные машины [23]), то теперь у него есть доступ внутрь вашей сети.

Это – одна из причин, почему до сих пор используются honeypot низкой и средней интерактивности: чем проще механизм honeypot, тем меньше возможностей для уязвимостей. Помимо этого, использование honeypot не должно быть слишком затратным: многие хосты не могут позволить себе запускать новую виртуальную машину на каждую атаку, и, если хакер заметит это, то он может запустить много соединений, получить много виртуальных машин, и тем самым запустить DoS-атаку на ваш хост.

Тем не менее, даже honeypot средней и низкой интерактивности представляют небольшой уровень риска. В самом базовом смысле, это еще одна программа, к которой имеется доступ извне, и следовательно еще один компонент, за безопасностью которого следует следить (хотя, в случае низко-интерактивных honeypot, возможностей для уязвимостей ограниченное количество, по большей части вокруг используемой SSH-библиотеки). Может быть полезно использовать honeypot в комбинации с другими системами безопасности, вроде fail2ban.

В некоторых SSH-клиентах есть поддержка port knocking [12]: чтобы подключиться к серверу, сначала выполняется “секретный стук”, который состоит из попыток подключиться к определенным закрытым портам. Если клиент успешно выполнил секретный стук, только тогда для него открывается доступ к SSH-серверу. Такой механизм легко объединить с honeypot: если не выполнен секретный стук, то по порту для SSH доступен honeypot-сервер, а если выполнен, то настоящий сервер [1]. (Следует учесть, что секретный стук – это не надежный механизм защиты, потому что любой, кто прослушивает сеть, может увидеть последовательность подключений и повторить ее. Настоящий сервер также должен иметь надежную аутентификацию, но с таким механизмом она не будет встречаться с большинством попыток фоновой радиации.)

5 Противодействия

Хакеры не хотят попасться на honeypot, поэтому они разрабатывают методы обнаружения их [7]. Один из методов – сравнение строк версий. Каждый SSH-сервер при установлении TCP-подключения пишет строку версии. Мы не можем банить IP-адреса просто из-за того, что они попробовали установить TCP-соединение, или даже за то что они попробовали залогиниться в систему – может быть, кто-то случайно напечатал свой IP-адрес неправильно, или же это просто один из сканеров всего интернета вроде Shodan.io (хотя самый высокий уровень паранойи предлагает заблокировать и их тоже).

Хакеры могут посмотреть на эту строку версии и пытаются понять, настоящий ли сервер или honeypot. Большинство honeypot притворяются частыми продуктами – вроде OpenSSH – но, в отличие от настоящих серверов, которые получают регулярные обновления, эти являются нестандартной частью установки, про которую можно забыть. Например, если только что вышло обновление, исправляющее критическую уязвимость в OpenSSH версии 10.0, и на всех серверах в этой организации теперь версия 10.1, а на этом все еще 10.0, то это хороший индикатор, что это на самом деле honeypot. Аналогично, если сервер возвращает одну строку ответа, а ведет себя на уровне пакетов как-то несвойственно этой версии, то это также индикатор того, что он не настоящий.

Если хакер уже смог подключиться и залогиниться в honeypot, то он уже спалился по IP-адресу. Но попытки взлома чаще всего делают с эвфемерных IP-адресов: либо принадлежащих VPN-службам, либо устройствам в ботнетах. Поэтому потерять один IP-адрес для него не слишком важно, потому что еще найдутся сотни взломанных компьютеров (или умных лампочек), с которых можно продолжить атаку по этой цели.

Гораздо опаснее для хакера – подключиться к высоко- или средне-интерактивному honeypot, и не заметить это, и попытаться использовать ценный эксплойт. Чрезвычайно ценны эксплойты «Zero-Day» – такие, о которых нет публичной информации.

Рынок распространения таких эксплойтов измеряется в миллионах долларов [16], и в нем участвуют организации вроде американского National Security Agency [21]. Это потому, что эксплойты, про которые никто не знает, позволяют взломать даже такие цели, которые имеют последние обновления безопасности.

На первый взгляд, такой эксплойт стоит использовать как можно больше, чтобы тем самым взломать весь мир в одночасье. Однако, если хакер выберет как цель honeypot-сервер с наблюдением, то он оставит для кибербезопасников очень много улик. Как только они разберут тот код, который хакер попытался запустить там, они смогут довольно быстро сотрудничать с разработчиками уязвимого ПО, чтобы выпустить обновление и тем самым очень сильно понизить ценность этого эксплойта.

Следовательно, подключившись к honeypot-серверу, хакеры могут оценить правдопод своего окружения: наличие и хеш-суммы системных файлов, возможность выполнять определенные редкие команды, наличие и свойства аппаратных устройств, возможность доступа к сети и сетевых соседей [24]. По этим характеристикам можно определить, насколько похож этот компьютер на настоящий (а также просто определить, насколько он представляет интерес в целом), и по результатам такого анализа хакеры могут принимать решение, насколько ценный эксплойт можно здесь использовать.

6 Заключение

Мы рассмотрели здесь только SSH-honeypot, но есть много разных типов. Они имеют подобные категории – от низкоинтерактивных, которые просто внешне выглядят как настоящий сервер, до высокинтерактивных, которые представляют собой изолированную копию настоящей системы. Они используются как компонент defence-in-depth, и не могут быть использованы сами по себе как полноценное решение безопасности; сами по себе, без других механизмов, они на самом деле делают вашу безопасность хуже, и только в сочетании с firewall это делает безопасность лучше.

Несмотря на всю пользу honeypot, нельзя забывать про традиционные способы защиты SSH и других сервисов, вроде отключения входа по паролю. Не следует надеяться, что плохие люди настолько будут отвлечены вашим фейковым SSH-сервером, чтобы не заметить настоящего SSH-сервера рядом с ним – особенно потому что большинство плохих людей не люди.

7 Список литературы

1. *Arifianto R. M.*, *Sukarno P.*, *Jadied E. M.* An SSH Honeypot Architecture Using Port Knocking and Intrusion Detection System // 2018 6th International Conference on Information and Communication Technology (ICoICT). — 2018. — С. 409—415. — DOI: 10.1109/ICoICT.2018.8528787.
2. *buixor*. CrowdSec. — 2020-2023. — URL: https://github.com/crowdsecurity/crowdsec ; [Online; accessed 7-декабрь-2023].
3. *Department for Digital, Culture, Media & Sport*. Cyber Security Breaches Survey 2022. — 2022. — URL: https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2022/cyber-security-breaches-survey-2022 ; [Online; accessed 7-декабрь-2023].
4. *desaster*. kippo. — 2011-2023. — URL: https://github.com/desaster/kippo ; [Online; accessed 7-декабрь-2023].
5. *Fahrnberger G.* Realtime Risk Monitoring of SSH Brute Force Attacks //. — 06.2022. — С. 75—95. — ISBN 978-3-031-06667-2. — DOI: 10.1007/978-3-031-06668-9_8.
6. *Fisher D.* Scanning the Internet in 45 Minutes. — 2013. — URL: https://threatpost.com/scanning-the-internet-in-45-minutes/102025/ ; [Online; accessed 7-декабрь-2023].
7. Identification of SSH Honeypots Using Machine Learning Techniques Based on Multi-Fingerprinting / Y.-J. Zhang [и др.] // 2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC). Т. 6. — 2023. — С. 1376—1381. — DOI: 10.1109/ITNEC56291.2023.10082467.
8. *janosdebugs*. containerssh. — 2020-2023. — URL: https://github.com/containerssh/containerssh ; [Online; accessed 7-декабрь-2023].
9. *JustinAzoff*. ssh-auth-logger. — 2016-2022. — URL: https://github.com/JustinAzoff/ssh-auth-logger ; [Online; accessed 7-декабрь-2023].

10. *Kaspersky Lab.* What is a honeypot? — 2023. — URL: https://www.kaspersky.com/resource-center/threats/what-is-a-honeypot ; [Online; accessed 7-декабрь-2023].

11. *McCune R.* CVE-2022-0185 in Linux Kernel Can Allow Container Escape in Kubernetes. — 2022. — URL: https://blog.aquasec.com/cve-2022-0185-linux-kernel-container-escape-in-kubernetes ; [Online; accessed 7-декабрь-2023].

12. *McKay D.* How to Use Port Knocking on Linux (and Why You Shouldn't). — 2019. — URL: https://www.howtogeek.com/442733/how-to-use-port-knocking-on-linux-and-why-you-shouldnt/ ; [Online; accessed 7-декабрь-2023].

13. *micheloosterhof.* cowrie. — 2015-2023. — URL: https://github.com/cowrie/cowrie ; [Online; accessed 7-декабрь-2023].

14. *micheloosterhof.* victpork. — 2018-2019. — URL: https://github.com/victpork/sshsyrup ; [Online; accessed 7-декабрь-2023].

15. *ohmyadd.* wetland. — 2017-2023. — URL: https://github.com/ohmyadd/wetland ; [Online; accessed 7-декабрь-2023].

16. *Perlroth N.* The Untold History of America's Zero-Day Market. — 2021. — URL: https://www.wired.com/story/untold-history-americas-zero-day-market/ ; [Online; accessed 7-декабрь-2023].

17. *Ribeiro P., Domanski R.* CVE-2021-27245. — 2021. — URL: https://www.cve.org/CVERecord?id=CVE-2021-27245 ; [Online; accessed 7-декабрь-2023].

18. *sebres.* fail2ban. — 2006-2023. — URL: https://github.com/fail2ban/fail2ban ; [Online; accessed 7-декабрь-2023].

19. *sjinks.* ssh-honeypotd. — 2014-2023. — URL: https://github.com/sjinks/ssh-honeypotd ; [Online; accessed 7-декабрь-2023].

20. *stephank.* lazyssh. — 2020-2021. — URL: https://github.com/containerssh/containerssh ; [Online; accessed 7-декабрь-2023].

21. *Tung L.* NSA: Our zero days put you at risk, but we do what we like with them. — 2014. — URL: https://www.zdnet.com/article/nsa-our-zero-days-put-you-at-risk-but-we-do-what-we-like-with-them/ ; [Online; accessed 7-декабрь-2023].

22. *Wikipedia contributors.* Internet background noise — Wikipedia, The Free Encyclopedia. — 2023. — URL: https://en.wikipedia.org/w/index.php?title=Internet_background_noise&oldid=1188601229 ; [Online; accessed 7-декабрь-2023].

23. *Wojtczuk R.* CVE-2012-0217. — 2012. — URL: https://www.cve.org/CVERecord?id=CVE-2012-0217 ; [Online; accessed 7-декабрь-2023].

24. *Неизвестный автор.* 20100316-233121-1847.log. — 2010. — URL: http://kippo.rpg.fi/playlog/?l=20100316-233121-1847.log ; [Online; accessed 7-декабрь-2023].