ИСПОЛЬЗОВАНИЕ DNS-ТУННЕЛИРОВАНИЯ ДЛЯ ПЕРЕДАЧИ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Москвичев А.Д.[1], Москвичева К.С.[2]

Цель статьи: *разработать способ увеличения уровня защищенности информационной системы от атаки с использованием DNS-туннелирования.*

Метод: *использование энтропии для выявления доменов и поддоменов, использующихся при передаче данных через DNS-туннель.*

Полученный результат: *рассмотрен метод передачи данных через протокол DNS в обход средств защиты информации. Осуществлена передача вредоносного файла с использованием DNS-туннелирования, произведен анализ работы средств защиты информации при передаче. Средства защиты информации не выявляют передачу вредоносного файла через DNS-протокол, однако выявляют в случае передачи в открытом виде. Дано понятие энтропии информации, ее роли в обработке данных. С помощью расчета энтропии для доменных имен выявлен домен, используемый при передаче вредоносного файла по DNS-туннелю. Сделан вывод о том, что энтропия может использоваться не только для выявления передачи данных через DNS-туннель, но и выявления активности вредоносного программного обеспечения, использующего в своей работе случайные доменные и поддоменные имена.*

Научная новизна *заключается в том, что вредоносная активность выявляется без использования базы знаний. Нет необходимости сигнатурно проверять каждый DNS запрос, достаточно рассчитать энтропию для выявления атаки.*

Ключевые слова: *компьютерная атака, защита информации, suricata, энтропия, SIEM, брокер сообщений, elasticsearch.*

DOI:10.21681/2311-3456-2022-4-91-99

Введение

DNS (англ. Domain Name System – система доменных имён) – компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене (SRV-запись). Распределённая база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определённому протоколу. Архитектура DNS остается неизменной с момента своего создания, но функции существенно изменились. Начиная с 90- х годов, DNS стали использовать в качестве инструмента балансировки нагрузки серверов информационных ресурсов – алгоритм Round Robin, который применяется серверами доменных имен при ответе на запросы клиентов. Роль доменного имени в процессе установки соединения – получение IP адреса.

DNS представляет собой службу стека TCP/IP, преобразующую доменные имена в IP-адрес или, наоборот, конвертирующую IP-адрес в компьютерное или доменное имя. Этот процесс называется разрешением имен или адресов[3].

1. DNS протокол, общие сведения

Система DNS функционирует по схеме рекурсивных или итеративных запросов, когда компьютер-клиент обращается за IP-адресом указанного пользователем доменного имени к DNS-серверу, явно указан-

3 Мищенко П.В. Сетевые службы FTP и DNS: учебное пособие / П.В. Мищенко. – Новосибирск : Новосибирский государственный технический университет, 2018. – 66 с. – ISBN 978-5-7782-3731-5.

1 Москвичев Антон Дмитриевич, аспирант, ФГБОУ ВО «Тихоокеанский государственный университет», г. Хабаровск, Россия. E-mail: anton.moskvichev.1996@yandex.ru, ORCID: 0000-0001-6532-2463

2 Москвичева Ксения Сергеевна, студент, ФГБОУ ВО «Тихоокеанский государственный университет», г. Хабаровск, Россия. E-mail: 2016104073@pnu.edu.ru.

ному в свойствах его подключения к компьютерной сети или сообщаемому провайдером. В результате поиска DNS-сервер должен разрешить, то есть преобразовать доменное имя в IP-адрес и вернуть его клиенту. При итеративном методе разрешения имён DNS-сервер выступает в роли клиента и опрашивает другие DNS-сервера в порядке убывания, начиная от корневых DNS-серверов и заканчивая последним, авторитетным за нужную DNS-зону [4]. Рассмотрим, как работает данный метод:

1. Пользователь хочет получить доступ по имени www.inadmin.ru и отправляет запрос на свой DNS-сервер.
2. DNS сервер видит, что пришёл запрос, и у него в кэше нет ответа, какой IP-адрес у www.inadmin.ru.
3. Так как сервер не знает, где находится www.inadmin.ru, он обращается к корневому DNS-серверу, и спрашивает, где находится www.inadmin.ru.
4. Корневой DNS-сервер не знает, где хранятся записи для домена www.inadmin.ru, но знает, кто ответственный за домен первого уровня .ru и возвращает нашему DNS серверу его IP, например, 193.232.142.17.
5. Наш DNS сервер обращается к 193.232.142.17 с просьбой сообщить IP для www.inadmin.ru. Но этот DNS тоже не знает ничего про наш адрес. Но знает, что есть DNS-сервер, который отвечает за inadmin.ru ивозвращает его IP, например, 195.128.64.3.
6. Наш DNS сервер обращается к 195.128.64.3 с просьбой сообщить IP для www.inadmin.ru. А вот этот сервер уже имеет нужную нам ресурсную запись, в которой указан IP-адрес, который мы ищем, и возвращает его нашему DNS-серверу.
7. Наш DNS сервер возвращает данный IP-адрес клиенту. Теперь клиент может подключиться по имени к серверу www.inadmin.ru.

При рекурсивном методе DNS-сервер просто пересылает данные другому DNS-серверу, чтобы тот выполнил всю работу (рекурсивно или итеративно) и вернул искомые данные, то есть возлагает задачу «хождения» по авторитетным DNS-серверам на своего «коллегу».

Кроме того, существует прямой и обратный DNS-запрос. Система DNS преобразовывает имена в IP-адреса и обратно. Обратное преобразование и осуществляется по обратному DNS-запросу. Для этого зарезервирован специальный домен in-addr.arpa, в котором хранятся PTR-записи. Октеты IP адреса хранятся в обратном порядке. Так для ip 1.2.3.4 будет создана запись вида 4.3.2.1.in-addr.arpa.

При запросе имени происходит несколько важных процедур, которые необходимо учитывать. Во-первых, данные о связке имя – IP адрес может храниться в нескольких местах (Hosts, DNS Cash, Lmhosts, DNS Server и др). Для того что бы полностью понимать принцип работы – нужно знать порядок, в котором Windows пытается разрешить любое имя.

1. При разрешении имени сверяется с локальным именем компьютера.
2. Если локальное имя не совпадает с запрашиваемым, то выполнятеся поиск в DNS Cash. В DNS-кэш динамически загружаются данные из файла HOSTS, поэтому поиск по файлу hosts не происходит, его данные всегда в памяти ПК, что ускоряет обработку. Файл Hosts расположен в%systemroot% \System32\Drivers\Etc.
3. Если имя не разрешилось в IP адрес, то пересылается на DNS сервер, который задан в сетевых настройках.
4. Если имя сервера плоское (к примеру: server1) и не может быть разрешено с помощью DNS, то имя конвертируется в NetBIOS имя и ищется в локальном кэше.
5. Если имя не может разрешиться, то ищется на WINS серверах.
6. Если имя не может быть определено и на WINS сервере, то ищется с помощью BROADCAST запроса в локальной подсети.
7. Если имя не определилось, то ищется в файле LMHOSTS.

Поиск по всем 7-ми шагам прекращается, как только находится первое вхождение, удовлетворяющие условиям. Посмотреть кэш можно по команде ipconfig /displaydns. Очистить кэш можно по команде ipconfig/flushdns.

Ресурсная запись является главной структурной единицей системы DNS, с помощью которой система выполняет свои функции. Фактически в ресурсных записях DNS-сервера содержатся все сведения, которые необходимы, чтобы дать ответ на поступивший DNS-запрос. Рассмотрим основные виды ресурсных записей [5].

— Запись A (address) – это главная адресная запись, необходимая для связи домена и IP-

4 Основы администрирования информационных систем : учебное пособие : [16+] / Д. О. Бобынцев, А. Л. Марухленко, Л. О. Марухленко [и др.]. – Москва ; Берлин : Директ-Медиа, 2021. – 201 с. : ил., табл.

5 Сергеев, А. Н. Основы локальных компьютерных сетей : учебное пособие для вузов / А. Н. Сергеев. – 3 е изд., стер. – Санкт-Петербург : Лань, 2021. – 184 с. – ISBN 978-5-8114-6855-3.

адреса сервера. Проще говоря, для работы сайта и всех поддоменов. Для протокола IPv4 используется запись A, для протокола IPv6 – запись AAAA.

— CNAME (Canonical name) – каноническое имя для псевдонима. Запись CNAME чаще всего используется для переадресации поддомена на другой домен.
— MX (Mail Exchanger) – адрес почтового шлюза для домена. Состоит из двух частей – приоритета и адреса узла. Записи MX критически важны для работы почты. Благодаря им отправляющая сторона «понимает» на какой сервер нужно отправлять почту для вашего домена.
— NS (Authoritative name server) – адрес узла, отвечающего за доменную зону. Проще говоря, запись NS указывает, какие DNS-серверы хранят информацию о домене. Критически важна для работы службы DNS.
— Обратная DNS-запись PTR связывает IP-адрес сервера с его каноническим именем (доменом). PTR-запись широко применяется в фильтрации почты. Для всех серверов виртуального хостинга REG.RU обратные DNS-записи уже прописаны. Если у вас виртуальный сервер VPS или выделенный сервер, прописать PTR-запись можно по инструкции.
— SOA (Start of Authority) – указывает, на каком сервере хранится эталонная информация о доменном имени. Критически важна для работы службы DNS.
— SPF (Sender Policy Framework) – указывает сервера, которые могут отправлять почту от имени домена. Запись SPF вносят в TXT-запись домена.
— TXT (Text string) – содержит любую текстовую запись. Широко применяется для проверок на право владения доменом при подключении дополнительных сервисов, а также для записи SPF и ключа DKIM. Записей TXT может быть сколько угодно.

2. Передача зараженного файла методом DNS-туннелирования

Пусть злоумышленник получил доступ к узлу в информационной системе. Ему необходимо доставить вредоносное программное обеспечение на узел. Однако, на периметре информационной системы установлен межсетевой экран, способный анализировать трафик на уровне приложения. Поэтому использование нелегитимных протоколов может скомпрометировать его нахождение в информационной системе.

Поставленная задача решается путем передачи вредоносного программного обеспечения методом DNS-туннелирования. Алгоритм действий:

1. Злоумышленник размещает в сети подконтрольный себе DNS-сервер.
2. Злоумышленник кодирует вредоносное программное обеспечение в текстовый формат, например, строку base64.
3. Злоумышленник размещает полученную строку на DNS-сервере в качестве ресурсной записи типа TXT. Для большей маскировки можно разбить строку на подстроки и разместить в соответствие разным поддоменам.
4. Злоумышленник на узле с помощью утилиты nslookup запрашивает ресурсную запись у DNS-сервера, тем самым получает вредоносное программное обеспечение в формате base64, преобразует с помощью powershell строку base64 в исполняемый файл.

Для реализации атаки злоумышленник использует DNS-сервер bind9 с настройками для обслуживания домена evil.local [1, с.184; 2, с.383]. Информация о поддоменах домена evil.local расположена в файле /etc/bind/db.evil.local.

Для проведения эксперимента разработан сценарий на языке bash (рис. 1) [3, с.187] для настройки DNS-сервера для передачи зараженного файла. В качестве зараженного файла выбрано вредоносное программное обеспечение mimikatz, представляющее собой бинарный файл [4, с.136; 5, с.363]. Сценарий преобразует файл mimikatz в формат base46 и записывает результат в файл mimi.b64. Далее для каждой строки из mimi.b64 сценарий генерирует имя поддомена и записывает в базу вместе с соответствующей строкой. В качестве имени поддомена используется хеш MD5. Дополнительно сценарий записывает полученные имена поддомена в файл urls. Этот файл нужен для сборки файла mimikatz на стороне жертвы, однако в нем нет необходимости если доменные имена можно сгенерировать на стороне жертвы средствами операционной системы.

Получение файла через DNS-туннель осуществляется с помощью сценария на языке powershell (рис. 2) [6, с.23]. Сценарий принимает сгенерированный список поддоменов. Для каждого имени поддомена делает запрос к DNS-серверу злоумышленника, тем самым получая части файла mimikatz в формате base64, и записывает в файл mimi.b64. Далее

```
base64 mimikatz.exe > mimi.b64
i=1
while read p;
do
link=`echo sig-$i | md5sum | awk '{print $1}'` ;
echo "$link.evil.local. IN TXT $p" >> /etc/bind/db.evil.local ;
echo "$link.evil.local" >> ./urls ;
let i+=1 ;
done < mimi.b64
```

Рис. 1. Сценарий для настройки DNS-сервера для передачи зараженного файла

```
$file=Get-Content -Path "urls"
foreach ($l in $file) {
$res=Resolve-DnsName -Type TXT $l -Server 192.168.253.128
$res | Select-Object -Property Type -ExpandProperty Strings >> mimi.b64
}
$file=Get-Content -Path "mimi.b64"
[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($file)) > mimikatz.exe
```

Рис. 2. Сценарий, получающий файл mimikatz через DNS-туннель

```
05/02/2022-06:52:59.103554 [**] [1:2016141:7]
    ET INFO Executable Download from dotted-quad Host [**]
    [Classification: A Network Trojan was detected]
    [Priority: 1] {TCP} 192.168.253.1:50914 -> 192.168.253.128:80
05/02/2022-06:52:59.103582 [**] [1:2034636:2]
    ET INFO Python SimpleHTTP ServerBanner [**]
    [Classification: Misc activity]
    [Priority: 3] {TCP} 192.168.253.128:80 -> 192.168.253.1:50914
05/02/2022-06:52:59.106993 [**] [1:2018959:4]
    ET POLICY PE EXE or DLL Windows file download HTTP [**]
    [Classification: Potential Corporate Privacy Violation]
    [Priority: 1] {TCP} 192.168.253.128:80 -> 192.168.253.1:50914
05/02/2022-06:52:59.106993 [**] [1:2021076:2]
    ET HUNTING SUSPICIOUS Dotted Quad Host MZ Response [**]
    [Classification: Potentially Bad Traffic]
    [Priority: 2] {TCP} 192.168.253.128:80 -> 192.168.253.1:50914
05/02/2022-06:52:59.139748 [**] [1:2029335:2]
    ET MALWARE Mimikatz x64 Executable Download Over HTTP [**]
    [Classification: A Network Trojan was detected]
    [Priority: 1] {TCP} 192.168.253.128:80 -> 192.168.253.1:50914
05/02/2022-06:53:22.398185 [**] [1:2034636:2]
    ET INFO Python SimpleHTTP ServerBanner [**]
    [Classification: Misc activity]
    [Priority: 3] {TCP} 192.168.253.128:80 -> 192.168.253.1:50985
05/02/2022-06:53:22.474513 [**] [1:2034636:2]
    ET INFO Python SimpleHTTP ServerBanner [**]
    [Classification: Misc activity]
    [Priority: 3] {TCP} 192.168.253.128:80 -> 192.168.253.1:50986
```

Рис. 3. Журналы системы обнаружения вторжений suricata при передаче файла mimikatz

```
05/02/2022-06:17:33.542144 [**] [1:2027027:2]
    ET ATTACK_RESPONSE UTF8 base64 string /This Program/ in DNS TXT Reponse [**]
    [Classification: Potentially Bad Traffic]
    [Priority: 2] {UDP} 192.168.253.128:53 -> 192.168.253.1:63618
05/02/2022-06:17:33.542144 [**] [1:2027030:2]
    ET ATTACK_RESPONSE UTF16-LE base64 string /This Program/ in DNS TXT Reponse [**]
    [Classification: Potentially Bad Traffic]
    [Priority: 2] {UDP} 192.168.253.128:53 -> 192.168.253.1:63618
```

Рис. 4. Журналы системы обнаружения вторжений suricata при передаче файла mimikatz через DNS-туннель

сценарий декодирует содержимое файла mimi.b64 и записывает в файл mimikatz.exe, который является исполняемым файлом mimikatz.

Пусть в информационной системе, где находится компьютер жертвы, работает система обнаружения вторжений suricata. Тогда в случае передачи вредоносного программного обеспечения по открытому каналу, например, через протокол HTTP, система обнаружения вторжений видит вредоносные сигнатуры и оповещает отдел информационной безопасности с помощью соответствующих записей в журналах (рис. 3) [7, с.135].

В случае, когда вредоносное программное обеспечение передано через DNS-туннель, система обнаружения вторжений suricata не наблюдает вредоносных сигнатур (рис. 4). Сообщение от системы обнаружения вторжений связано с запросом к DNS-серверу типа TXT, однако это штатная работа системы, такое сообщение, вероятнее всего, отдел ИБ проигнорирует.

3. Обнаружение DNS-туннелирования с помощью расчета энтропии

Рассмотрим случайную величину ξ, которая принимает значения $x_1,\ldots,x_N$ с вероятностями $p_1,\ldots,.p_N$. Возникает вопрос, как можно количественно охарактеризовать связь между априорной информацией о случайной величине ξ и ее функцией распределения. Эта связь должна отражать меру неопределенности нашего знания о случайной величине ξ. Такая мера неопределенности была введена Шенноном в виде выражения:

$$H_N = -\sum_{i=1}^{N} p_i log p_i. \qquad (1)$$

Выражение (1) это энтропия распределения вероятностей

$$P_N = \left\{ (p_1,\ldots,p_N) \in R^N : \; p_i \geq 0, p_1 + \ldots + p_N = 1 \right\}. \qquad (2)$$

Если о случайной величине ξ ничего не известно, то из принципа максимума энтропии следует, что максимум энтропии достигается при $p_i = N^{-1}$, при этом $H_{max} = logN$, что совпадает с качественными представлениями о неопределенности (более размытое распределение имеет большую неопределенность, чем распределение с явно выраженным пиком). Энтропия распределения вероятностей характеризует неопределенность исходов опыта, трудность предсказания его результата [8, с.311].

На практике при расчете энтропии для строк имеется закономерность: чем выше случайность строки, тем выше энтропия. Так как имена поддоменов, использующиеся злоумышленником при передаче данных методом DNS-туннелирования, имеют случайный характер, то их энтропия должна быть наибольшей.

Пусть имеется 500 легитимных DNS-имен и одно DNS-имя, используемое при передаче данных методом DNS-туннелирования. Для расчета энтропии разработан скрипт на языке программирования python (рис. 5) [9, с.421].

Таблица 1 содержит 10 доменов с наибольшей энтропией, рассчитанной сценарием. Самая высокая энтропия оказалась у домена, участвующего в передаче данных методом DNS-туннелирования.

4. Использование энтропии в SIEM-системах

Для успешной реализации мероприятий защиты информационных инфраструктур необходимо решить ряд задач, связанных с созданием системы мониторинга угроз безопасности. Системы мониторинга реализуют апостериорный подход к защите информации, главная цель создания которой – снижение внутреннего и внешнего воздействий на инфраструктурные объекты до минимального уровня риска и минимизация возникающего ущерба [10, с.9].

Одним из наиболее перспективных и эффективных направлений в создании систем мониторинга угроз безопасности в настоящее время считается использо-

```
import math

def entropy(string):
    prob = [ float(string.count(c)) / len(string) for c in dict.fromkeys(list(string)) ]
    entropy = - sum([ p * math.log(p) / math.log(2.0) for p in prob ])
    return entropy

def entropy_ideal(length):
    prob = 1.0 / length
    return -1.0 * length * prob * math.log(prob) / math.log(2.0)

f=open("domains.txt", "r")
for d in f:
    d=d[:-1]
    e=str(entropy(d)).replace('.', ',')
    print("{}\t{}".format(d, e))
```

Рис. 5. Сценарий расчета энтропии для доменов

Таблица 1

Результаты вычисления энтропии для доменов

Доменное имя	Энтропия	Доменное имя	Энтропия
be8115a828faa92f99b55d8 77836eb00.evil.local	4,138262941	wrestling.com	3,700439718
antispyware.com	3,773557262	partysupplies.com	3,690116518
1stbandwidth.com	3,75	forsalebyowner. com	3,683542362
marketingtoday.com	3,725480557	university.com	3,664497779
healthinsurance.com	3,721611724	giftbasket.com	3,664497779

вание SIEM-системы, обеспечивающей управление информацией и событиями безопасности.

Учитывая характер и содержание задач защиты в сервисных и критических инфраструктурах, представляется целесообразным положить в основу построения системы мониторинга концепцию SIEM-системы.

Основная цель построения и функционирования SIEM-систем –значительно повысить уровень информационной безопасности в информационной инфраструктуре за счет обеспечения возможности манипулировать информацией о безопасности и осуществлять проактивное управление инцидентами и событиями безопасности в режиме близком к реальному времени.

Проактивный означает «действующий до того, как ситуация станет критической». Предполагается, что проактивное управление инцидентами и событиями безопасности основывается на автоматических механизмах, которые используют информацию об «истории» анализируемых сетевых событий и прогнозе будущих событий, а также на автоматической подстройке параметров мониторинга событий к текущему состоянию защищаемой системы.

Пусть имеется абстрактный SIEM (рис. 6), который получает информацию о DNS запросах, производимых из информационной системы. Информация поступает от источника событий в виде записей из журналов. Источником событий может выступать межсетевой экран, системы обнаружения и предотвращения вторжений и иные. Записи из журналов попадают в нормализатор, который преобразует их к единому формату и передают брокеру в качестве сообщений. Сообщения забирают из брокера различные модули для дальнейшей обработки.

Использование энтропии для выявления передачи данных через DNS-туннель реализуется двумя способами. Первый, программное обеспечения подключается к брокеру сообщений в качестве модуля и, получая нормализованные события, делает расчет энтропии. В качестве брокера могут выступать RabbitMQ [11, с.175], Apache Kafka [12, с.93], NATS [13, с.131] и другие. Такой способ наиболее эффек-

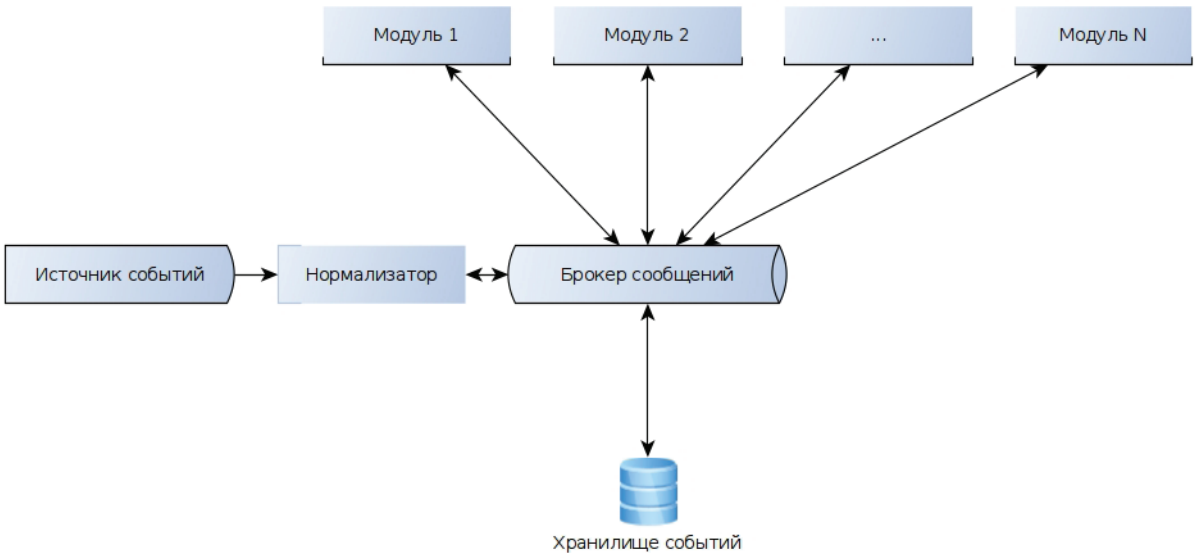

Рис. 6. Общая архитектура SIEM-систем

тивный, однако невозможен в случае использования стороннего SIEM, так как требует изменения архитектуры. Второй, программное обеспечения с заданной периодичностью подключается к хранилищу событий и делает расчеты. Хранилищем могут выступать Neo4j [14, с.164], Elasticsearch [15, с.81], Yandex ClickHouse и другие. Такая реализация проще и подходит для любого SIEM, однако не так эффективна с точки зрения времени реагирования в силу большой задержки.

Вывод

Использование энтропии помогает обнаружить не только DNS-туннели, но и вредоносное программное обеспечение и веб-эксплойты, использующие домены и поддомены, созданные с использованием алгоритма генерации доменов. Злоумышленники используют алгоритмы генерации доменов для создания случайно выглядящих доменов и поддоменов, используя своего рода «ключ» или «соль», которые могут расшифровать только они. Затем эти домены можно использовать для будущих атак. Поскольку эти домены генерируются случайным образом и могут работать только в течение короткого промежутка времени, специалистам по защите информации чрезвычайно сложно заблокировать их с помощью традиционных методов, таких как черные списки.

Использование программного обеспечения, вычисляющего энтропию, в качестве модуля SIEM позволяет быстро и эффективно выявлять DNS-имена, свидетельствующие о вредоносной активности, не используя базу знаний. Однако необходимо помнить, что энтропия – величина относительная. Слишком низкий порог будет приводить к частым ложным срабатываниям, а высокий – к пропускам.

Литература

1. Левицкий Н. Д. Удаленный сервер своими руками. От азов создания до практической работы: руководство / Н. Д. Левицкий. – Санкт-Петербург: Наука и Техника, 2021. – 400 с. – ISBN 978-5-94387-568-7.
2. Колисниченко Д. Н. LINUX. Полное руководство по работе и администрированию: руководство / Д. Н. Колисниченко. – Санкт-Петербург: Наука и Техника, 2021. – 480 с. – ISBN 978-5-94387-608-0.
3. Донцов, В. П. Linux на примерах: руководство / В. П. Донцов, И. В. Сафин. – Санкт-Петербург : Наука и Техника, 2017. – 352 с. – ISBN 978-5-94387-742-1.
4. Диогенес, Ю. Кибербезопасность. стратегия атак и обороны / Ю. Диогенес, Э. Озкайя; перевод с английского Д. А. Беликова. – Москва: ДМК Пресс, 2020. – 326 с. – ISBN 978-5-97060-709-1.
5. Белоус А. И. Основы кибербезопасности. Стандарты, концепции, методы и средства обеспечения: энциклопедия / А. И. Белоус, В. А. Солодуха. – Москва : Техносфера, 2021. – 482 с. – ISBN 978-5-94836-612-8.
6. Бертрам А. Powershell для сисадминов / А. Бертрам – Санкт-Петербург: Издательский дом «Питер», 2021 – 416 с.
7. Коллинз, Майкл. Защита сетей. Подход на основе анализа данных / Майкл Коллинз. – М.: ДМК Пресс, 2020. – 307 с.: ил. – ISBN 978-5-97060-649-0.
8. Авдошин С.М. Дискретная математика. Модулярная алгебра, криптография, кодирование / С.М. Авдошин, А.А. Набебин. – Москва: ДМК Пресс, 2017. – 352 с. – ISBN 978-5-94074-408-3.

9. Борзунов С. В. Алгебра и геометрия с примерами на Python / С. В. Борзунов, С. Д. Кургалин. – 3-е изд., стер. – Санкт-Петербург : Лань, 2022. – 444 с. – ISBN 978-5-8114-9980-9.
10. Абденов А.Ж., Трушин В.А., Сулайман К. Анализ, описание и оценка функциональных узлов SIEM-системы [Книга]. – Новосибирск: Новосибирский государственный технический университет, 2018. – стр. 122.
11. Gavin M. Roy RabbitMQ in Depth / Gavin M. Roy — Shelter Island, NY: Manning Publications, 2018. — 264 p.
12. Narkhede N., Shapira G., Palino T. Kafka: The Definitive Guide. Real-Time Data and Stream Processing at Scale / Neha Narkhede, Gwen Shapira, Todd Palino — Sebastopol, USA: O'Reilly Media, 2017. — 566 p.
13. Quevedo W. Practical NATS / Waldemar Quevedo — San Francisco, California, USA: APRESS, 2018. — 260 p.
14. Нидхем М., Ходлер Э. Графовые алгоритмы. Практическая реализация на платформах Apache Spark и Neo4j. / пер. с англ. В. С. Яценкова – М.: ДМК Пресс, 2020. – 258 с.
15. Sachdeva G. S. Practical ELK Stack: Build Actionable Insights and Business Metrics Using the Combined Power of Elasticsearch, Logstash, and Kibana / Gurpreet S. Sachdeva — San Francisco, California, USA: APRESS, 2017. — 318 p.

USING DNS TUNNELING TO TRANSFER MALICIOUS SOFTWARE

Moskvichev A.D.[6], Moskvicheva K.S.[7]

Purpose of the article: *to develop a way to increase the level of protection of an information system from an attack using DNS tunneling.*

Method: *using entropy to identify domains and subdomains used when transferring data through a DNS tunnel.*

The result: *a method of data transmission through the DNS protocol bypassing the information security tools is considered. A malicious file was transferred using DNS tunneling, and an analysis was made of the operation of information protection tools during transmission. Information security tools do not detect the transfer of a malicious file via the DNS protocol, but they do if it is transferred in clear text. The concept of information entropy, its role in data processing is given. By calculating the entropy for domain names, the domain used in the transmission of a malicious file through the DNS tunnel was identified. It is concluded that entropy can be used not only to detect data transfer through the DNS tunnel, but also to detect the activity of malicious software that uses random domain and subdomain names in its work.*

The scientific novelty *lies in the fact that malicious activity is detected without using the knowledge base. There is no need to signature check each DNS request, it is enough to calculate the entropy to detect an attack.*

Keywords: *computer attack, information protection, suricata, entropy, SIEM, message broker, elasticsearch.*

References

1. Levickij, N. D. Udalennyj server svoimi rukami. Ot azov sozdaniya do prakticheskoj raboty : rukovodstvo / N. D. Levickij. – Sankt-Peterburg: Nauka i Tekhnika, 2021. – 400 s. – ISBN 978-5-94387-568-7.
2. Kolisnichenko, D. N. LINUX. Polnoe rukovodstvo Po rabote i administrirovaniyu : rukovodstvo / D. N. Kolisnichenko. – Sankt-Peterburg: Nauka i Tekhnika, 2021. – 480 s. – ISBN 978-5-94387-608-0.
3. Doncov, V. P. Linux na primerah : rukovodstvo / V. P. Doncov, I. V. Safin. – Sankt-Peterburg: Nauka i Tekhnika, 2017. – 352 s. – ISBN 978-5-94387-742-1.
4. Diogenes, YU. Kiberbezopasnost'. strategiya atak i oborony / YU. Diogenes, E. Ozkajya; perevod s anglijskogo D. A. Belikova. – Moskva : DMK Press, 2020. – 326 s. – ISBN 978-5-97060-709-1.
5. Belous, A. I. Osnovy kiberbezopasnosti. Ctandarty, koncepcii, metody i sredstva obespecheniya: enciklopediya / A. I. Belous, V. A. Soloduha. – Moskva: Tekhnosfera, 2021. – 482 s. – ISBN 978-5-94836-612-8.
6. Bertram A. Powershell dlya sisadminov / A. Bertram – Sankt-Peterburg: Izdatel'skij dom «Piter», 2021 – 416 s.

6 Anton D. Moskvichev, postgraduate, Pacific National University, Khabarovsk, Russia. E-mail: anton.moskvichev.1996@yandex.ru. ORCID: 0000-0001-6532-2463

7 Ksenia S. Moskvicheva, student, Pacific National University, Khabarovsk, Russia. E-mail: 2016104073@pnu.edu.ru.

7. Kollinz, Majkl. Zashchita setej. Podhod na osnove analiza dannyh / Majkl Kollinz. – M.: DMK Press, 2020. – 307 s.: il. – ISBN 978-5-97060-649-0.
8. Avdoshin, S.M. Diskretnaya matematika. Modulyarnaya algebra, kriptografiya, kodirovanie / S.M. Avdoshin, A.A. Nabebin. – Moskva: DMK Press, 2017. – 352 s. – ISBN 978-5-94074-408-3.
9. Borzunov, S. V. Algebra i geometriya s primerami na Python / S. V. Borzunov, S. D. Kurgalin. – 3-e izd., ster. – Sankt-Peterburg : Lan', 2022. – 444 s. – ISBN 978-5-8114-9980-9.
10. Abdenov A.ZH., Trushin V.A., Sulajman K. Analiz, opisanie i ocenka funkcional'nyh uzlov SIEM-sistemy [Kniga]. – Novosibirsk: Novosibirskij gosudarstvennyj tekh-nicheskij universitet, 2018. – str. 122.
11. Gavin M. Roy RabbitMQ in Depth / Gavin M. Roy — Shelter Island, NY: Manning Publications, 2018. — 264 p.
12. Narkhede N., Shapira G., Palino T. Kafka: The Definitive Guide. Real-Time Data and Stream Processing at Scale / Neha Narkhede, Gwen Shapira, Todd Palino — Sebastopol, USA: O'Reilly Media, 2017. — 566 p.
13. Quevedo W. Practical NATS / Waldemar Quevedo — San Francisco, California, USA: APRESS, 2018. — 260 p.
14. Nidhem M., Hodler E. Grafovye algoritmy. Prakticheskaya realizaciya na platformah Apache Spark i Neo4j. / per. s angl. V. S. YAcenkova – M.: DMK Press, 2020. – 258 s.
15. Sachdeva G. S. Practical ELK Stack: Build Actionable Insights and Business Metrics Using the Combined Power of Elasticsearch, Logstash, and Kibana / Gurpreet S. Sachdeva — San Francisco, California, USA: APRESS, 2017. — 318 p.