

Отчет анализа дампа Серов Глеб Ильич

1. Участвующие хосты

В дампе присутствуют следующие основные узлы:

IP-адрес	Роль
192.168.118.129	Клиент
192.168.118.128	Сервер
192.168.118.1	Предполагаемый шлюз (ответов не наблюдается в дампе)

Сеть относится к диапазону 192.168.118.0/24.

2. Общая картина взаимодействия.

Взаимодействие между клиентом и сервером происходит в несколько этапов:

1. ARP-резолвинг (клиент пытается определить MAC-адреса других узлов).
2. HTTP-соединение (порт 1337) – передача данных в открытом виде.
3. TLS-соединение (порт 1338) – защищенный канал.

3. Анализ переданных данных.

ARP-трафик

В дампе наблюдаются ARP-запросы вида “Who has 192.168.118.1?/Who has 192.168.118.128?”. Ответы получены только от сервера 192.168.118.128 (рис. 1).

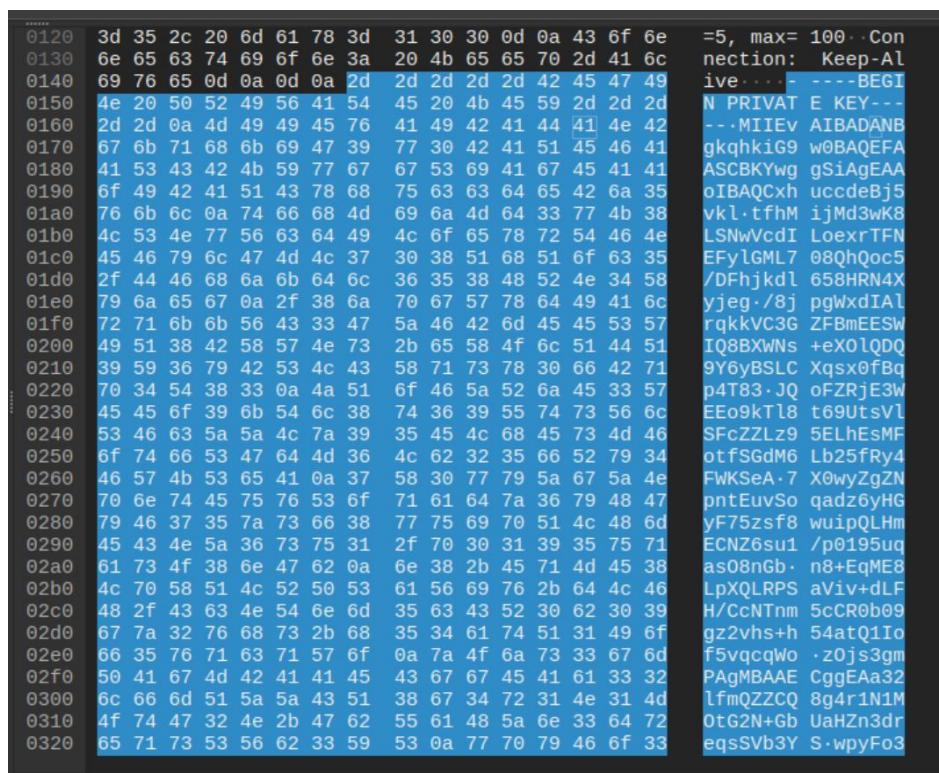
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
2	3.005975	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
3	4.004544	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
4	5.004453	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
5	8.009009	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
6	9.008399	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.1? Tell 192.168.118.129
7	9.509763	VMware_23:75:b6	Broadcast	ARP	42	Who has 192.168.118.128? Tell 192.168.118.129
8	9.511210	VMware_ff:a1:64	VMware_23:75:b6	ARP	60	192.168.118.128 is at 00:0c:29:ff:a1:64

Рисунок 1 – Ответ от сервера на ARP запрос.

В дампе не наблюдается защиты ARP (по типу ARP ACL, DAI), то есть, в сети отсутствуют механизмы предотвращения ARP-spoofing.

HTTP-трафик (порт 1337).

Клиент выполняет HTTP-запрос “GET /k3y HTTP/1.1”. Ответ сервера содержит закрытый криптографический ключ в формате PEM (рис. 2).



The screenshot shows a hex dump of a large amount of data. The data starts with several lines of ASCII text, which is the PEM header for an RSA private key:

```
-----  
-----BEGIN PRIVATE KEY-----  
-----END PRIVATE KEY-----
```

Following this header, there is a large block of binary data representing the RSA private key's modulus and exponent. The data is presented in a standard hex dump format, where each byte is shown as a two-digit hexadecimal value followed by its ASCII representation.

Рисунок 2 - Ответ сервера

Данный ключ передается без шифрования, что является критической утечкой секретной информации.

TLS-трафик (порт 1338).

После получения ключа клиент устанавливает TLS-соединение с сервером. Можно проследить следующий используемый параметр (рис. 3): Cipher Suite (TLS_RSA_WITH_AES_256_CBC_SHA) – используется RSA key exchange, отсутствует Perfect Forward Secrecy (что достигается с помощью

DHE/ECDHE). То есть, RSA применяется для шифрования premaster secret, который генерируется клиентом, шифруется публичным ключом сервера, расшифровывается сервером приватным ключом, и далее из premaster secret вычисляются master secret, session keys. То есть, безопасность сессии зависит от приватного ключа сервера.

```

- Transport Layer Security
  - TLSv1 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 68
  - Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 64
    Version: TLS 1.0 (0x0301)
    > Random: c6d0eaa...c6893
    Session ID Length: 0
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Compression Method: null (0)
    Extensions Length: 24
    > Extension: renegotiation_info (len=1)
    > Extension: session_ticket (len=0)
    > Extension: application_layer_protocol_negotiation (len=11)
      [JA3S Fullstring: 769,53,65281-35-16]
      [JA3S: bdb56d136854096c1c01af1fdbcc6893]
  - TLSv1 Record Layer: Handshake Protocol: Certificate
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 1014
  - Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 1010
    Certificates Length: 1007
    > Certificates (1007 bytes)
  - TLSv1 Record Layer: Handshake Protocol: Server Hello Done
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 4
  - Handshake Protocol: Server Hello Done
    Handshake Type: Server Hello Done (14)
  ● Cipher Suite (tls.handshake.ciphersuite), 2 bytes

```

Рисунок 3 – Использованный Cipher Suite

Далее возникла идея расшифровать TLS. Был скопирован полученный ранее ключ, проверен на валидность (рис. 4).

```

> openssl pkey -in server.key -check
Key is valid
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBqkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQCxhuccdeBj5vkl
tfhMijMd3wK8LSNwVcdILoexrTFNEFylGML708QhQoc5/Dfhjkdl658HRN4Xyjeg
/8jpgWxdIAlrqkkVC3ZFBmEESWIQ8BXWNs+eXOlDQ9Y6yBSLCXqsx0fBqp4T83
JQoFZRjE3WEe9kTl8t69UtsvlSLfcZLz95ElhEsMFotfSGdM6Lb25fRy4FWKSeA
7X0wyZgZNpntEuvSoqadz6yHGYF75zsfs8wuipQLHmECNZ6su1/p0195uqas08nGb
n8+EqME8LpxQLRPSaViv+dLFH/CcNTnm5cCR0b09gz2vhsh54atQ1Io5vqcqWo
z0js3gmPAgMBAAECCggEAa32lfmQZCQ8g4r1N1M0tG2N+GbUaHZN3dreqsSVb3YS
wpyFo3mL7h+cRY3gUJOxw6yxc/NNDtaQbFm0Sz7kYFa/nYNUEPwG4ot7D65aLXHo
inrw9ZkQXuTLPMiXcQiUdKEkfvuO1YKThk5VDQMzxi/NSMP08pocy0yxqH11W/5y
s8PwI2jAC7osJ4TUqNVNZezeNLm3EcmbXjYphpyRnMnVxIucUoEX9sXk+zLh7G
kYPAttIKfMZdwvTD1qsd8t8ANwKjd9aNhfhrsvIjv5C3WcsGj5eINwwainNb70GM
/mxaVuPaIm7PE0JU8mQrss2MvAeG/NuhTUCvTgmfoQKBgQDYRyfRM6atTgKKfJVZ
ySkBtb07DukcKb/dd6ls13PK08Xr7X/FTii3B1abPSBrpN9+v1wm46JG0ismrPcf
rm9m1NLvRLJEznVdGKuCQ9bR3x0uRBSkfTCMldxcn57/PIoxtLVHTVOXLIuw6W2Y
f/FCakn10IRZW5XPRYkzpkJcxwKBgQDSIcX/n9GzFiy6bfkvXFg2HUBch+NkB8su
jFIu9wgg0Zx1v5+0QH5Pl3Twh/Nb78cTnBgSmSMCDZmf2Tld4g/axj5V0RaEcNxA

```

Рисунок 4 – Проверка ключа на валидность

Далее в Wireshark был расшифрован TLS (рис. 5)

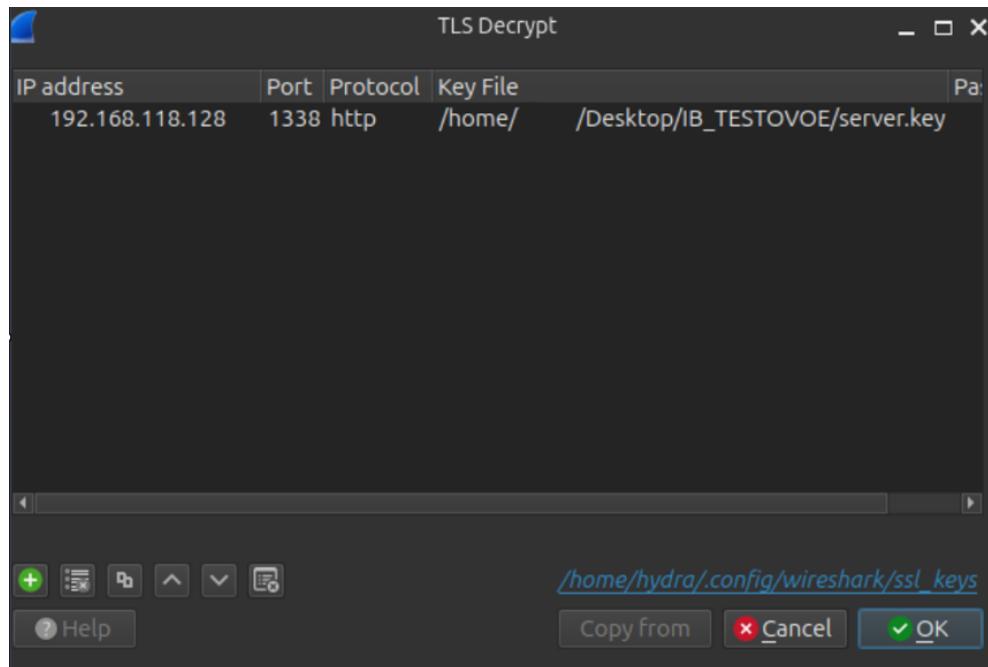


Рисунок 5 – Расшифровка TLS

В результате расшифрования обнаружен еще один GET запрос (рис. 6).

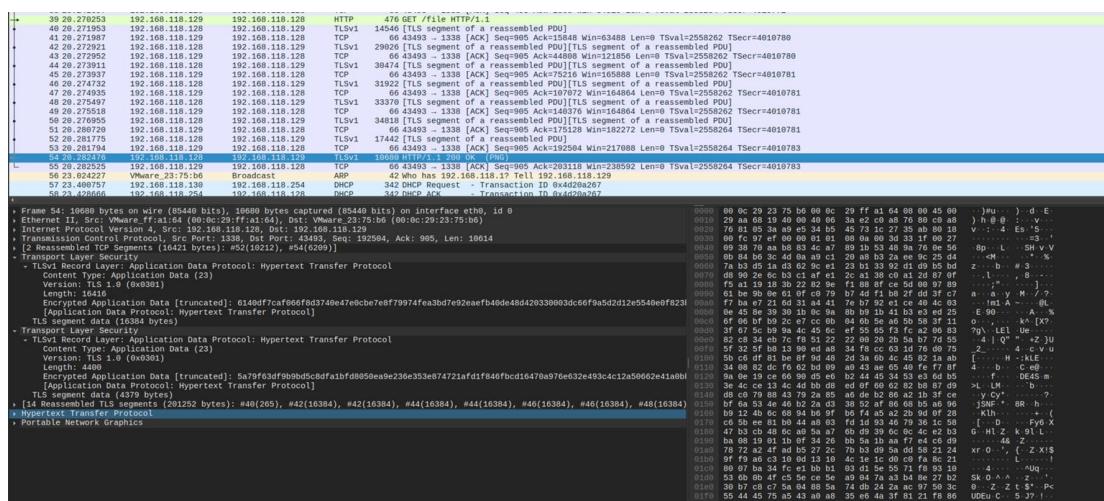


Рисунок 6 – GET-request

В ответе – PNG файл (рис. 7).

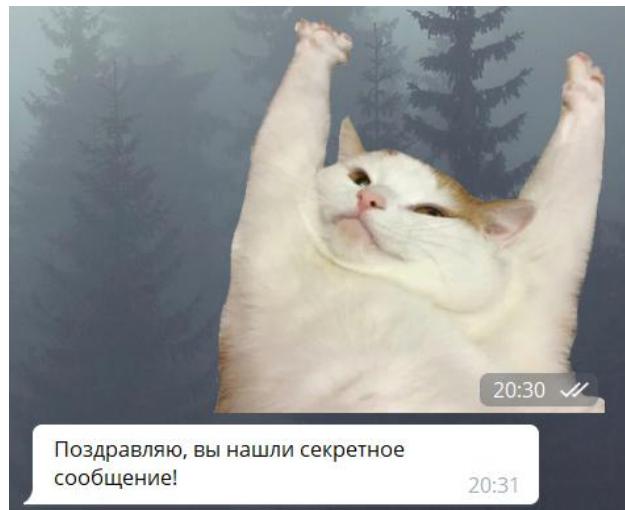


Рисунок 7 – Полученный PNG файл

5. Итоговые выводы

В дампе зафиксировано клиент-серверное взаимодействие. Передавались ARP-requests, HTTP-requests, закрытый RSA-ключ, TLS-трафик, бинарный PNG-файл. В результате произошла утечка закрытого ключа, TLS-трафик был расшифрован.

Из уязвимостей можно отметить отсутствие защиты от MITM в локальной сети (ARP без механизмов защиты). Закрытый ключ передается по HTTP – наиболее критичная уязвимость. Закрытый RSA-ключ передается в открытом виде, компрометируется вся модель доверия TLS. Также используется устаревший Cipher Suite.

Также HTTP-headers раскрывают тип сервера и версию ПО, что потенциально облегчает подбор известных уязвимостей.