Code:

```
import requests
from datetime import datetime, timedelta
from sqlalchemy import Integer, create_engine, Column, String, DateTime
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker

Base = declarative_base()

class Links(Base):
    __tablename__ = 'repos'
    url = Column(String, primary_key=True)
    last_push_time = Column(DateTime)

class RulesDB(Base):
    __tablename__ = 'rules'
    id = Column(Integer, autoincrement=True,primary_key=True)
    content = Column(String)

# SQLite database setup using SQLAlchemy
DATABASE_URL = 'sqlite:///rules.db'
engine = create_engine(DATABASE_URL)
Session = sessionmaker(bind=engine)
session = Session()

# Function to download rules from URLs listed in repos.DB
def update_rules():
    print("Updating rules from repositories...")
    session.query(RulesDB).delete()
    session.commit()
    repos = session.query(Links).all()
    for repo in repos:
        response = requests.get(repo.url)
        if response.status_code == 200:
            rule_content = response.text.split("\n")
            for i in rule_content:
                if i[:5]=="alert":
                    rule_entry = RulesDB(content=i[:-1])
                    session.add(rule_entry)
                    session.commit()
```

```
            print(f"Downloaded and saved rules")

# Function to append rules to Suricata rule directory
def push_rules():
    print("Pushing rules to Suricata directories...")
    rules = session.query(RulesDB).all()
    with open("/var/lib/suricata/rules/local.rules", "a") as f:
        for rule in rules:
            f.write(rule.content+"\n")

# Check if last push was more than 5 hours ago
def check_last_push_time():
    last_rule =
session.query(Links).order_by(Links.last_push_time.desc()).first()
    if last_rule!=None and last_rule.last_push_time:
        if datetime.now() - last_rule.last_push_time > timedelta(hours=5):
            return True
    return False

# Main function for update and push logic
Base.metadata.create_all(engine)
a=session.query(Links).filter_by(url="https://sslbl.abuse.ch/blacklist/ssl
blacklist.rules").first()
if a==None:

session.add(Links(url="https://sslbl.abuse.ch/blacklist/sslblacklist.rules
",last_push_time=datetime.now()))
    session.commit()
    update_rules()
    push_rules()
if check_last_push_time():
        print("Last push was more than 5 hours ago. Updating and pushing
rules...")
        update_rules()
        push_rules()

a=session.query(Links).filter_by(url="https://sslbl.abuse.ch/blacklist/ssl
blacklist.rules").first()
        a.last_push_time=datetime.now()
else:
```

```
        print("Rules are up to date.")
```

Output:

```
firewall@firewall:~$ sudo python3 suricata.py
```

Running the Python code:

```
firewall@firewall:~$ ls
fwbuilder-5.1.0.3599.dmg  suricata.py
firewall@firewall:~$ sudo python3 suricata.py
/home/firewall/suricata.py:13: MovedIn20Warning: The ``declarative_base()`` function is now availabl
e as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: ht
tps://sqlalche.me/e/b8d9)
  Base = declarative_base()
Updating rules from repositories...
Downloaded and saved rules
Pushing rules to Suricata directories...
Rules are up to date.
firewall@firewall:~$ _
```

Checking the Database:

```
firewall@firewall:~$ sqlite3 rules.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> select * from repos;
https://sslbl.abuse.ch/blacklist/sslblacklist.rules|2024-09-10 16:36:59.000183
sqlite> _
```

The repos Table has the link to the site having rules.

Rules Database:

```
sqlite> select * from rules;_
```

```
5876|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (D
arkGate C&C)"; tls.fingerprint:"9a:57:37:99:49:f7:34:b1:1f:32:bb:84:62:db:1f:3f:d8:89:87:22"; refere
nce:url, sslbl.abuse.ch/ssl-certificates/sha1/9a57379949f734b11f32bb8462db1f3fd8898722/; sid:9022058
75; rev:1;)
5877|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (C
obaltStrike C&C)"; tls.fingerprint:"6b:ef:20:79:08:bf:ad:6b:19:58:00:67:ce:77:0b:c8:20:d3:d2:ef"; re
ference:url, sslbl.abuse.ch/ssl-certificates/sha1/6bef207908bfad6b19580067ce770bc820d3d2ef/; sid:902
205876; rev:1;)
5878|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (A
syncRAT C&C)"; tls.fingerprint:"be:4d:4f:07:7b:dd:90:61:83:67:eb:2a:b8:8f:9f:30:74:cc:b7:aa"; refere
nce:url, sslbl.abuse.ch/ssl-certificates/sha1/be4d4f077bdd90618367eb2ab88f9f3074ccb7aa/; sid:9022058
77; rev:1;)
5879|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (D
CRat C&C)"; tls.fingerprint:"b7:85:d1:a9:e5:78:47:03:b9:8a:96:69:8a:d0:5d:ff:5e:07:22:9a"; reference
:url, sslbl.abuse.ch/ssl-certificates/sha1/b785d1a9e5784703b98a96698ad05dff5e07229a/; sid:902205878;
 rev:1;)
5880|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (R
hadamanthys C&C)"; tls.fingerprint:"b3:31:52:6a:09:49:f8:8c:e2:18:55:5e:df:10:60:c4:a0:2d:e5:a2"; re
ference:url, sslbl.abuse.ch/ssl-certificates/sha1/b331526a0949f88ce218555edf1060c4a02de5a2/; sid:902
205879; rev:1;)
5881|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (R
hadamanthys C&C)"; tls.fingerprint:"4d:a1:42:24:45:2c:1f:e6:1f:46:b1:11:2c:43:ec:fd:9f:32:2c:82"; re
ference:url, sslbl.abuse.ch/ssl-certificates/sha1/4da14224452c1fe61f46b1112c43ecfd9f322c82/; sid:902
205880; rev:1;)
5882|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (D
CRat C&C)"; tls.fingerprint:"2c:39:32:73:7f:3e:e8:2f:a4:19:23:17:2d:e8:16:77:3a:89:87:15"; reference
:url, sslbl.abuse.ch/ssl-certificates/sha1/2c3932737f3ee82fa41923172de816773a898715/; sid:902205881;
 rev:1;)
5883|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (L
atrodectus C&C)"; tls.fingerprint:"7c:d8:0b:c6:f1:e5:a5:56:82:66:08:70:db:52:11:98:25:3e:ae:f9"; ref
erence:url, sslbl.abuse.ch/ssl-certificates/sha1/7cd80bc6f1e5a55682660870db521198253eaef9/; sid:9022
05882; rev:1;)
5884|alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (L
atrodectus C&C)"; tls.fingerprint:"88:37:19:f4:45:36:fc:17:fe:83:26:c9:26:9f:cc:cf:6b:d1:c1:80"; ref
erence:url, sslbl.abuse.ch/ssl-certificates/sha1/883719f44536fc17fe8326c9269fcccf6bd1c180/; sid:9022
05883; rev:1;)
sqlite>
```

Trying to run the python code within 5 hours:

```
firewall@firewall:~$ sudo python3 suricata.py
/home/firewall/suricata.py:13: MovedIn20Warning: The ``declarative_base()`` function is now availabl
e as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: ht
tps://sqlalche.me/e/b8d9)
  Base = declarative_base()
Rules are up to date.
firewall@firewall:~$
```

Checking suricata Rules:

```
root@firewall:/var/lib/suricata/rules# cat local.rules
```

```
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (DarkGa
te C&C)"; tls.fingerprint:"9a:57:37:99:49:f7:34:b1:1f:32:bb:84:62:db:1f:3f:d8:89:87:22"; reference:u
rl, sslbl.abuse.ch/ssl-certificates/sha1/9a57379949f734b11f32bb8462db1f3fd8898722/; sid:902205875; r
ev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (Cobalt
Strike C&C)"; tls.fingerprint:"6b:ef:20:79:08:bf:ad:6b:19:58:00:67:ce:77:0b:c8:20:d3:d2:ef"; referen
ce:url, sslbl.abuse.ch/ssl-certificates/sha1/6bef207908bfad6b19580067ce770bc820d3d2ef/; sid:90220587
6; rev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (AsyncR
AT C&C)"; tls.fingerprint:"be:4d:4f:07:7b:dd:90:61:83:67:eb:2a:b8:8f:9f:30:74:cc:b7:aa"; reference:u
rl, sslbl.abuse.ch/ssl-certificates/sha1/be4d4f077bdd90618367eb2ab88f9f3074ccb7aa/; sid:902205877; r
ev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (DCRat
C&C)"; tls.fingerprint:"b7:85:d1:a9:e5:78:47:03:b9:8a:96:69:8a:d0:5d:ff:5e:07:22:9a"; reference:url,
 sslbl.abuse.ch/ssl-certificates/sha1/b785d1a9e5784703b98a96698ad05dff5e07229a/; sid:902205878; rev:
1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (Rhadam
anthys C&C)"; tls.fingerprint:"b3:31:52:6a:09:49:f8:8c:e2:18:55:5e:df:10:60:c4:a0:2d:e5:a2"; referen
ce:url, sslbl.abuse.ch/ssl-certificates/sha1/b331526a0949f88ce218555edf1060c4a02de5a2/; sid:90220587
9; rev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (Rhadam
anthys C&C)"; tls.fingerprint:"4d:a1:42:24:45:2c:1f:e6:1f:46:b1:11:2c:43:ec:fd:9f:32:2c:82"; referen
ce:url, sslbl.abuse.ch/ssl-certificates/sha1/4da14224452c1fe61f46b1112c43ecfd9f322c82/; sid:90220588
0; rev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (DCRat
C&C)"; tls.fingerprint:"2c:39:32:73:7f:3e:e8:2f:a4:19:23:17:2d:e8:16:77:3a:89:87:15"; reference:url,
 sslbl.abuse.ch/ssl-certificates/sha1/2c3932737f3ee82fa41923172de816773a898715/; sid:902205881; rev:
1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (Latrod
ectus C&C)"; tls.fingerprint:"7c:d8:0b:c6:f1:e5:a5:56:82:66:08:70:db:52:11:98:25:3e:ae:f9"; referenc
e:url, sslbl.abuse.ch/ssl-certificates/sha1/7cd80bc6f1e5a55682660870db521198253eaef9/; sid:902205882
; rev:1;)
alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"SSLBL: Malicious SSL certificate detected (Latrod
ectus C&C)"; tls.fingerprint:"88:37:19:f4:45:36:fc:17:fe:83:26:c9:26:9f:cc:cf:6b:d1:c1:80"; referenc
e:url, sslbl.abuse.ch/ssl-certificates/sha1/883719f44536fc17fe8326c9269fcccf6bd1c180/; sid:902205883
; rev:1;)
root@firewall:/var/lib/suricata/rules# _
```

ls -l to see extra details about the file, link modified date etc..

```
root@firewall:/var/lib/suricata/rules# ls -l
total 40124
-rw-r--r-- 1 root root     3228 Jul 28 17:38 classification.config
-rw-r--r-- 1 root root 10818633 Sep 10 16:37 local.rules
-rw-r--r-- 1 root root 30260588 Jul 28 17:38 suricata.rules
root@firewall:/var/lib/suricata/rules#
```

The local.rules has been modified by our Python script.