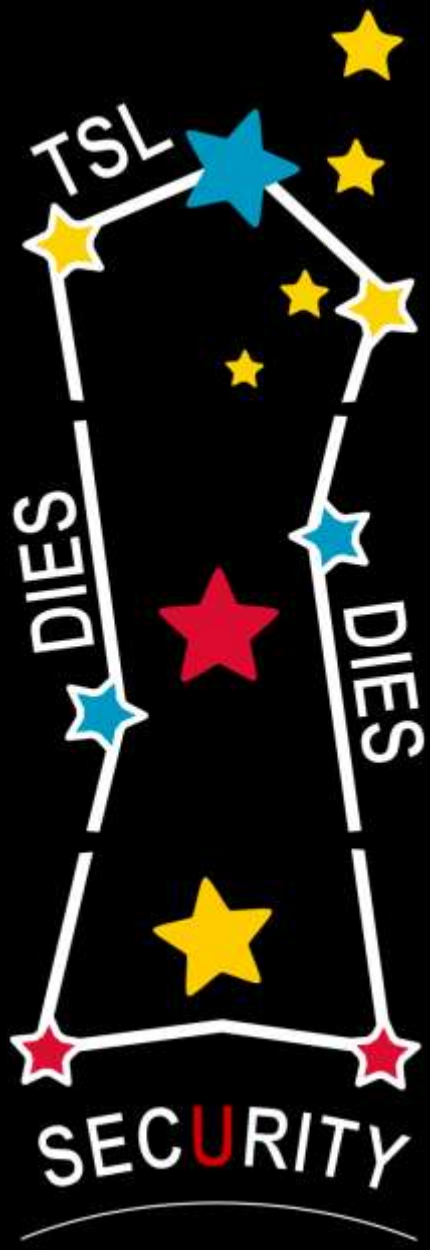
UNIVERSITY OF TWENTE.



DETECTING ZERO-DAY AND TARGETED ATTACKS AGAINST ICS

DINA HADZIOSMANOVIC

DAMIANO BOLZONI

DISTRIBUTED AND EMBEDDED SECURITY GROUP

THE CONTEXT

- Hermes, Castor and Midas
  - 3 projects sponsored by the (former) Dutch Ministry of Internal affairs

MOTIVATIONS: current countermeasures cannot detect the latest cyber threats against industrial control systems

- Stuxnet
- Vulnerabilities disclosed by “independent researchers”
- Project Basecamp

GOALS: enhance current approaches and develop new techniques

- Using data mining and anomaly detection techniques

PARTNERS

ABB

alliander

brabant Water

egemin AUTOMATION

FOX-IT EXPERTS IN IT SECURITY

gasunie

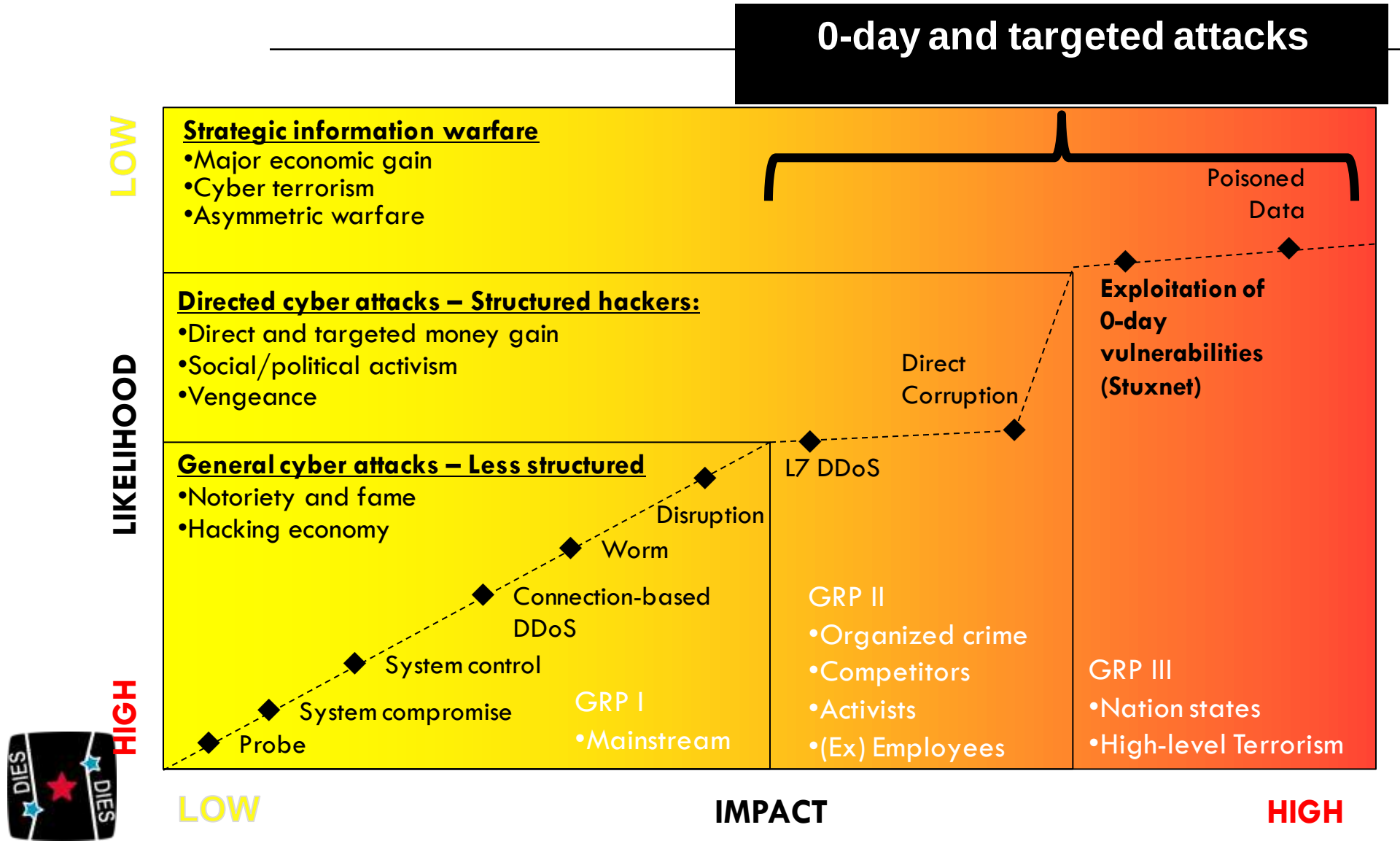
SECURITY MATTERS

waternet

UNIVERSITEIT TWENTE.

DIES DIES

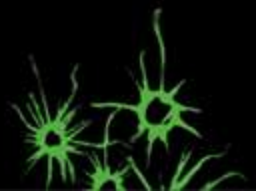
THE CYBER SECURITY PROBLEM



UNIVERSITY OF TWENTE.

HERMES

HOST-BASED EVENT MINING IN SCADA SYSTEMS



THE PROBLEM

- SCADA systems log thousands of events per day
  - User/system activities
- Logs are hardly analyzed/processed by operators
  - Too much work
  - Lack of skills
- A good deal of information is lost…

THREATS AND CURRENT SECURITY TOOLS

- NIDS/HIDS mainly address *system-related* threats
  - Buffer overflows
  - Virus/Worms
- What about:
  - Authorized users that make mistakes
  - Unauthorized users that gain enough privileges and perform malicious actions
- We call those “process-related threats”
  - Leverage vulnerabilities in the application logic
  - A higher semantic understanding of inputs is needed for detection

DETECTING PROCESS RELATED THREATS

- System logs provide a complete overview of the processes
  - We look for *rare* log entries
- Malicious/anomalous events are supposed to happen *rarely*
- Use visualization to ease the task of IT (security) operators
  - Support operators with little security skills

LOG NORMALIZATION

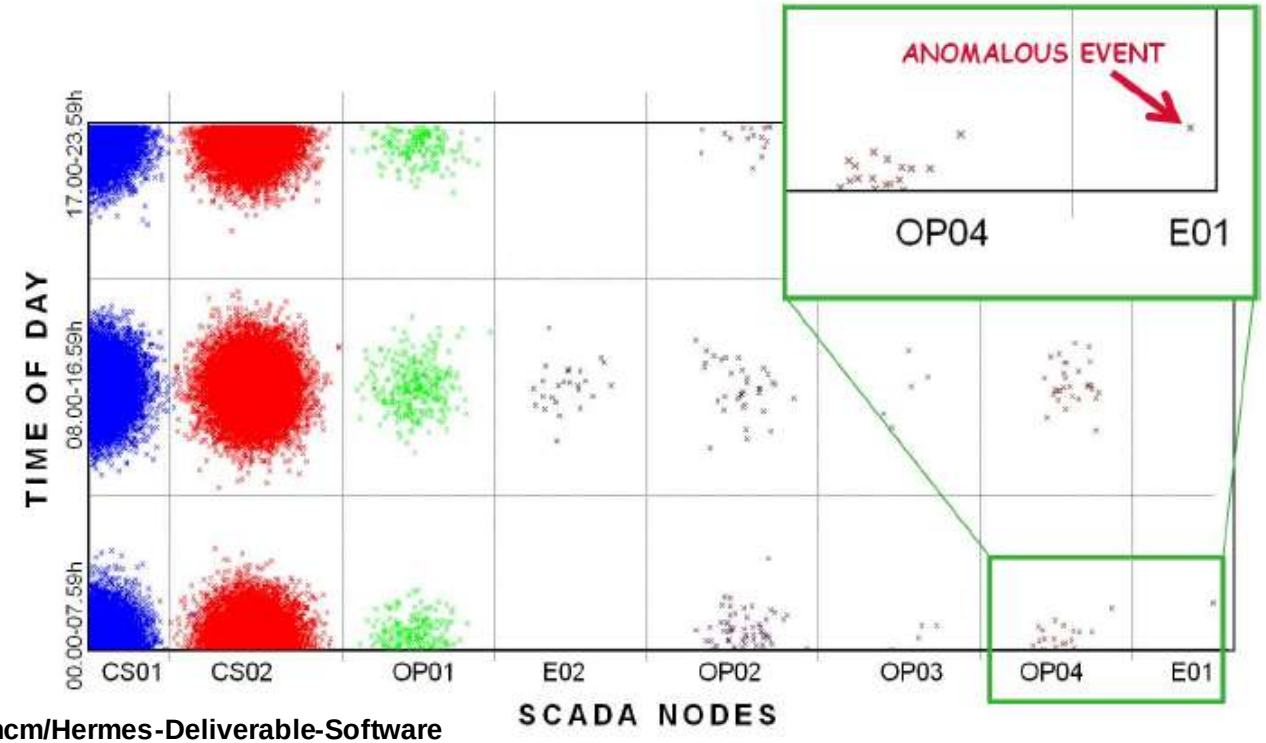
- A typical log entry
  - 31/07/2011 21:56:10,System Simple Event,Controller_Alg (2001) Interval time in ordinary tasks inc. 1.3,X.Y.Z.W-_SW1131Task,CSPAWPK01

- Each log entry has several attributes
  - Some are not relevant ("locale")
  - Some are incomplete ("user account")
  - Some require pre-processing ("timestamp" → working shift)

- Together with process engineers we selected the most "interesting" ones
  - *Timestamp (Working shift), SCADA node, Object_path, Type of event, Aspect of event* and *User account*

EXPERIMENTS

- We plot a graphical representation of the events
  - 14 days of logs, ~100K events
- No intrusion had been reported during the chosen days
  - But…

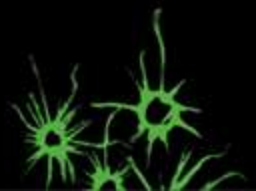


https://zeus.tsl.utwente.nl/wiki/hcm/Hermes-Deliverable-Software

UNIVERSITY OF TWENTE.

CASTOR

CONTROLLING ACCESS TO SCADA NETWORKED SYSTEMS



THE PROBLEM

- At some point in time, despite the organization's policies, an unauthorized device is connected to the network
  - A technician that needs to run some maintenance, perhaps with a malware-infected laptop
- A disgruntled employee could use his knowledge and trust level to plant a malware into some systems (e.g., an HMI client)

APPROACH

- Approach
  - Add seamlessly "smart" ACLs to current installations
    - Automatically build **a model of the network** that describes **communication patters and protocols** used among hosts
    - For some protocols, enforce **function codes** normally used
  - Communications with an **abnormal pattern** are flagged as anomalous

BENCHMARKS IN REAL-LIFE ENVIRONMENTS

CURRENT STATUS

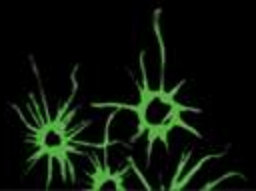
- The system has been deployed in a real-life production site
  - MMS, OPC and SMB
  - Training for **5 minutes**, 2 false alerts over 7 days of testing

- Then we re-deployed it in a testing environment
  - This environment was supposed to be a copy of the production site, actually it wasn't → **the system spotted the inconsistency**
  - We connected an unauthorized device → **detected**
  - We simulated a hacked authorized device using a different set of protocols/function codes → **detected**

UNIVERSITY OF TWENTE.

MIDAS

INTRUSION DETECTION FOR SCADA SYSTEMS



PROBLEM

- Current NIDS are mainly based on signatures
  - Blacklisting
  - Cannot detect 0-day exploitations, because they lack the proper signatures
  - Some implementations use heuristics to improve detection, but with little success

- Anomaly detection (whitelisting) has been advocated as the definite solution for years
  - So far, only flow-based anomaly detection systems managed to penetrate the market → cannot detect in general a data injection exploit
  - Too many false alerts in real-life environments

APPROACH

- Include a **(partial) specification** of the **protocol** to monitor
  - Lower false alerts, increases detection capabilities
- If a network message **is not protocol-compliant** an alert is raised
- The detection engine **"learns" normal values** for all of the protocol message fields
  - Numbers/Lengths: enumerations, ranges (for instance, $0 < x < 100$)
  - Strings: regular expressions
  - Binary buffers: byte frequency distribution
- Messages with **abnormal** field **values are** flagged as **attacks**

FIRST BENCHMARKS IN CONTROLLED ENVIRONMENT

CURRENT STATUS

- We use data sets collected at four production sites from project partners
  - ✓ Modbus tests
- Detects the RPC exploits used by Stuxnet
  - ✓ The system detects that the RPC functions exploited have not been seen before
  - ✓ We then simulated the use of the “NetprPathCompare” function (MS-08-067), and re-run the exploit -(too much data is sent compared to normal usage)
- Tested against Wurldtech’s Achilles
  - ❑ Modbus → all tests cleared with success

SUMMARY

- HERMES – detect legitimate but undesirable commands on the application level
- CASTOR - monitor your plant and derive models of communication
- MIDAS – monitor message fields and look for anomalous packets

QUESTIONS

?