Artificial Intelligence for cyber security

22-23 January 2019 / Lille/ France
Protection of an information system by an AI: a three-phase approach based on behaviour analysis to detect a hostile scenario

**INTRODUCTION & CONCLUSION**
Sylvain NAVERS

**APPROACH & RESULTS**
Jean-Philippe FAUVELLE
Alexandre DEY

1. Needs
2. SIEM solutions
3. UEBA concept
4. Our approach
5. POC #1: scenario
6. POC #1: behind the scene
7. POC #1: results
8. POC #1: conclusion
9. POC #2: scenario
10. POC #2: behind the scene
11. POC #2: results
12. POC #2: conclusion
13. Situation and future
14. Your questions
A. Real world: 4 cases to illustrate detection completeness and quality.

B. Needs.

1. Needs

- Detect hostile actions over wide time periods, including APT*. 
- Produce explainable alerts.
- Automatically adapt to changing threats and behaviors.
- Reduce false positives/negatives.
- Horizontal scaling.

### REAL WORLD
- Growing and evolving threats.
- Hostile actions over wide time periods, including APT*.
- Cyber and non-cyber events.
- Weak signals, noises, pollution.
- Increasing volume of data.

### MAIN NEEDS
- Detect hostile actions over wide time periods, including APT*.
- Produce explainable alerts.
- Automatically adapt to changing threats and behaviors.
- Reduce false positives/negatives.
- Horizontal scaling.

### Events chain is hostile?

<table>
<thead>
<tr>
<th>Case</th>
<th>Events chain spread over a wide time period</th>
<th>Correctness</th>
<th>Explainability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>True negative</td>
<td>Yes (complete)</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>True positive</td>
<td>Yes (complete)</td>
</tr>
<tr>
<td>4</td>
<td>Yes (APT*)</td>
<td>True positive</td>
<td>Yes (complete)</td>
</tr>
</tbody>
</table>

(*) APT – Advanced Persistent Threat
2. SIEM* solutions

A. SIEM pros and cons.
B. Four cases to show limits.

---

### Pros & Cons of current SIEM* solutions

- **Many mature solutions**
- **Widely used**
- **Weak signals difficult to detect**
- **Inter-event time relationships difficult to detect**
- **Static settings, to be maintained continuously**

---

#### Events chains spread over a wide time period

- **Case 1**: No
- **Case 2**: Yes
- **Case 3**: Yes
- **Case 4**: Yes (APT)

---

#### SIEM*

- **Correctness**
  - False positive
  - True positive
  - True posit. Two alerts
  - False negative

- **Explainability**
  - N/A
  - Yes
  - Yes (partial)
  - N/A

---

(*) SIEM – Security Information and Event Management
QUICK FACTS CONCERNING UEBA*  
- Learning of behaviours.  
- Method agnostic to Good/Evil: detects behaviour changes (incongruities).  
- Two training methods:  
  - Once for all training (eg: embarked).  
  - Continuous training: assimilation and forgetting of behaviours, permanent adaptation, non-supervised.  

**UEBA with continuous training meets our needs.**

MAIN BIASES OF AVAILABLE SOLUTIONS  
- Training performance.  
- Many false positives (or negatives).  
- Slightly explainable result (black box).  
- Over-simplification of problems to solve.  
- Almost systematic presence of a simple time window alerts counter.  
- Little consideration of events temporality.  
- Low management of behavioural model, boiled frog paradox (see below).

UEBA PRINCIPLE AND BOILED FROG PARADOX  
- Assimilate new behaviours:  
  - Need for **quick** synchronism.  
- Avoid boiled frog paradox:  
  - Need for **slow** synchronism.  

**Conflicting needs: synchronism is an unsatisfactory compromise.**

(*) UEBA – User and Entity Behaviour Analytics
### Approach
- POC #1 (finished): simulated activity on an information system (with synthetic data).
- POC #2 (almost finished): real activity on a workstation (with real data).
- Keep in mind biases.
- Focus on explainability of results.

### Principle
- Three phases AI:
  - Learning (continuous).
  - Inference.
  - Correlation.
- AI for memorisation (to be done).
Compromising documents on a company's information system, by screening / targeting, identity theft, malicious attachment, and exploitation of a vulnerability.

**Usual behaviours (extract)**

14 Normal sending of internal and external emails.
15 Normal solicitation of equipments / ports.
18 Normal activity between the external and the equipment compromised.

**Hostile scenario**

10 The hacker performs a screening and targeting.
11 The hacker prepares an attack kit.
12 The hacker sends an email with malicious attachment to 2 targeted employees by usurping a third-party identity.
13 The hacker sends an email with malicious attachment to 2 targeted employees by usurping a third-party identity.
14 The hacker prepares an attack kit.
15 The hacker sends an email with malicious attachment to 2 targeted employees by usurping a third-party identity.
16 Targeted employee opens the attachment and activates the charge.
17 The charge scans ports on vulnerable equipment and compromises one.
18 The hacker connects to the compromised equipment and takes control of it.
19 The hacker exploits the vulnerability to collect sensitive documents.
20 An OSINT* source reports hacker.

(*) OSINT – Open Source Intelligence
A company, 100 employees working on site and from their home.
- Theatre: an IS (internal/external PC, messaging, network flows, firewalls, routers).
- Internal, external, mixed flows.
- A social network used for screening / targeting.

Our own massive, coherent data generator.
- 500K metrics generated.
- Data enrichment (e.g., aggregations / counts on sliding time windows).

- Input metrics: converted to numbers.
- Algorithm: isolation forest, unsupervised.
- Output scores: neither normalised nor filtered, so that the correlation phase (see below) receives all the information including weak signals.
- Real time performance: ~5K metrics / s. on a single PC.

Discovery of major interest graphs, with an algorithm working on 3 spaces:
1. Metrics concentration (quasi-twins).
2. Search for related events.
3. Search for major interest graphs made of strong / weak / normal signals via a relevance function.

6. POC #1: behind the scene
- Scenario details.
- Metrics generation.
- More about AI.
- Correlation and graphs.
A. Achieved expected results.

- Few false positives (during calibration).

B. Unexpected results.

- Detection of suspicious flow: sending of the same malicious attachment to the employee’s PC n° 2.
- Detection of a fourth behavioural incongruity \( B_4 \): the hacker downloads sensitive documents located on PC n° 48.

Detection is complete with good explainability.
<table>
<thead>
<tr>
<th>Main Biases of Available Solutions</th>
<th>Our Results for POC #1</th>
<th>Focus (for POC #2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training performance</td>
<td>□ Learning: partially scalable.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Inference + correlation: horizontal scaling.</td>
<td></td>
</tr>
<tr>
<td>Many false positives</td>
<td>□ Few false positives, only during first month (calibration).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ No false negatives.</td>
<td></td>
</tr>
<tr>
<td>Over-simplification of problems</td>
<td>□ Training on the entire dataset.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Multivariate events of different types.</td>
<td></td>
</tr>
<tr>
<td>Slightly explainable result</td>
<td>□ Detection is complete with good explainability.</td>
<td></td>
</tr>
<tr>
<td>Frequent presence of a simple time window alerts counter</td>
<td>□ We don’t use counters but graphs on sliding and variable time windows over wide temporal ranges.</td>
<td></td>
</tr>
<tr>
<td>Little consideration of events temporality</td>
<td>□ Our algorithm uses events temporality, it adapts to any time scale, from microseconds to years.</td>
<td></td>
</tr>
<tr>
<td>Low management of behavioural model, boiled frog paradox</td>
<td>□ To be done, we will use AI for synchronisation of the behavioural model.</td>
<td></td>
</tr>
<tr>
<td>Other limitations</td>
<td>□ Synthetic data.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Simplistic scenario.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>□ Too little data.</td>
<td></td>
</tr>
</tbody>
</table>
On his Linux PC, a user unwisely executes a malicious script which downloads an exploit from the Web in order to use a kernel vulnerability to elevate its privileges.

### Usual behaviours

The user performs office tasks (eg: word processing, messaging, Internet browsing).

The user executes commands and scripts.

### Hostile scenario

1. The user executes a malicious script, via a BASH* command.
2, 3, 4 The malicious script downloads source code of an exploit from the web, via a WGET* command.
5, 6, 7 The malicious script compiles the exploit, via a GCC* command.
8, 9 The malicious script executes the compiled exploit, which tries to elevate its privileges using a vulnerability of the operating system kernel.

(*) - BASH: standard command for executing scripts.  
   - WGET: standard command for downloading files from the Web.  
   - GCC: standard command for compiling programming languages.
- **Real data.**
- **12 million metrics** (2 millions / day).
- Theatre: inside a PC.
- Metrics collected through standard auditing functions of operating system.
- 90% kernel primitives calls.

**Metrics**
- Unauthorised actions.
- Calls to functions/commands for modifying kernel/modules.
- Suspicious actions (eg: nmap, wget, tcpdump).
- Access to monitored files (eg: config., binaries, temp. files).
- Commands executed.
- Invocations of potentially dangerous kernel primitives.
- Credentials (eg: user, group).
- Context (eg: path, timestamp, parent process).

- **Input metrics**: conversion of categorical variables to numerical using probability of observing couples of values after observing others.
- **Algorithm**: deep learning autoencoder, unsupervised.
- **Regularisation**: dropout, noise addition, early stopping.
- **Output scores**: normalised, not filtered.
- **Real time** performance: ~2K metrics / s. on a single PC with GPU.

- **Discovery of major interest graphs**: same as for POC #1.
A. Achieved expected results.

11. POC #2: results

**Main Results**

- **Detection is complete with good explainability**:  
  - Execution of the BASH script (score 0.1).  
  - Execution of the WGET command (score 0.6).  
  - Three executions of the GCC command (score 0.29).  
  - Execution of the exploit (score 0.29).

- The BASH process has a low incongruity score, but it still contributes to the major interest graph because it connects other actions.

- Some false positives resulting from rare actions, which could be avoided by optimising training.

- No false negatives.
<table>
<thead>
<tr>
<th>MAIN BIASES OF AVAILABLE SOLUTIONS</th>
<th>OUR RESULTS FOR POC #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many false positives</td>
<td>Few false positives, but could be avoided.</td>
</tr>
<tr>
<td></td>
<td>No false negatives.</td>
</tr>
<tr>
<td>Over-simplification of problems</td>
<td>Training on the entire dataset, directly from raw logs.</td>
</tr>
<tr>
<td></td>
<td>Multivariate events of different types.</td>
</tr>
<tr>
<td>Slightly explainable result</td>
<td>Detection is complete with good explainability.</td>
</tr>
<tr>
<td>Frequent presence of a simple time window alerts counter</td>
<td>We don't use counters but graphs on sliding and variable time windows over wide temporal ranges.</td>
</tr>
<tr>
<td>Little consideration of events temporality</td>
<td>Our algorithm uses events temporality, it adapts to any time scale, from microseconds to years.</td>
</tr>
<tr>
<td>Low management of behavioural model, boiled frog paradox</td>
<td>To be done, we will use AI for synchronisation of the behavioural model.</td>
</tr>
</tbody>
</table>

Other limitations  |  Simplistic scenario. |
**SITUATION**

- Effective association of UEBA with correlation process.
- Good explainability of alerts.
- Few but avoidable false positives.
- Temporality taken into account from microseconds to years.
- Real time 3 phases algorithm + horizontal scaling.
- Integration issues partially addressed (ELK).

**Encouraging results.**

**Results confirmed in various contexts.**

**FUTURE**

- More realistic scenarios.
- Adversarial AI*.
- Memorisation AI*.
- Interoperation with SIEM.

(*) PhD thesis 2019 : « Continuous Model Learning for Anomaly Detection In the Presence of Highly Adaptative Cyberattacks ». 

13. Situation and future

A. Progress and limits.
B. Remaining work.
Questions (and answers !)