Proc. of the European Safety and Reliability Conference (ESREL 2026)
Edited by José C. Matos, Paulo B. Lourenço, Michael Beer, Daniel Oliveira,
Edoardo Patelli, Hélder S. Sousa, Rui A. Silva and Monica Santamaria
 Published by Research Publishing, Singapore.
doi: 10.3850/ESREL2026061419_esrel26-p26230-cd

Safety methods for cyber security: using statecharts for the analysis of a cyber attack

Claudia Picoco
EDF R&D Palaiseau, France. E-mail: claudia.picoco@edf.fr

Jingxuan Ma
EDF R&D Palaiseau, France. E-mail: jingxuan.ma@edf.fr

Valentin Rychkov
EDF R&D Palaiseau, France. E-mail: valentin.rychkov@edf.fr

The significant increase in cyberattacks targeting the OT domain since 2022 and the growing number of cybersecurity requirements applicable to critical infrastructure require to implement more effective and practical cybersecurity risk management. Cyberattacks can be a potential source of failure in safety systems, thereby impacting safety, availability, and the lifespan of critical infrastructure such as nuclear power plants. Operational safety has long provided methods to analyze and quantify the risk associated to differents scenarios raising from an initiating event. and tools can therefore be suitable for the analysis of a cyberattack scenarios. Boolean methods (fault trees and event trees) have been historically used in probabilistic risk and safety assessment. Dynamic probabilistic risk and safety assessment methods are developed to explicitly account for timings of event in the accidental scenarios. In this work, we use statecharts, a formalism recently adopted for dynamic probabilistic risk assessment, for the analysis of a cyber-attack. Current approaches used in cyber analysis are mostly IT-oriented. These methods do not allow for a graphical representation and probabilities are only accounted in a qualitative way. The statecharts are a graphical approach that allow to visualize the systems, and account explicitly for dynamic interactions and the inter-dependencies amongst different part of the system. Moreover, the approach allows to integrate probabilities for quantitative insights. We analysed the following cyber-attack: a user receives emails containing malicious links, the user, who may inadvertently click on such links, regularly accesses a high-performance computing system — the ultimate target of the attack. The model incorporates protective measures such as antivirus software, system updates, and firewalls. Risk importance measures are calculated to gain insights and assess the efficiency of different security barriers.

Keywords: Dynamic PSA, statecharts, cyber-security

1. Introduction

Cyber-security risk assessment has become increasingly important as modern systems grow in inter-connectivity and complexity. The mitigation of cyber-attack scenarios is a central aspect of cyber-security risk management. Various approaches have been developed to construct cyber-attack scenarios. Expert-defined scenarios, often inspired by past incidents documented in cyber-attack databases, are commonly used as a basis for cyber risk management activities.

Graph-based methods, such as attack trees Lallie et al. (2020), provide a more systematic framework for constructing cyber-attack scenarios and for evaluating the effectiveness of countermeasures. By drawing an analogy with probabilistic risk and safety assessment (PSA), expert-driven cyber-security approaches can be related to qualitative Failure Modes and Effects Analysis (FMEA), whereas attack trees correspond more closely to fault tree or event tree analyses.

Although Boolean approaches are widely used for risk assessment in different domains, methods that more accurately capture the dynamic aspects of system behaviour have been actively developed and applied Moulié et al. (2022); Blaszkiewicz et al. (2024); IRSN (2024); Vasilyev et al. (2021); Picoco et al. (2020); Rychkov and Picoco (2022); Rychkov et al. (2023a).

In contrast to Boolean approaches, dynamic probabilistic reliability and

risk assessment (DPRA) methods do not require risk-relevant scenarios to be defined *a priori*. Instead, scenarios are generated automatically as an outcome of the analysis capturing time dependent interaction between different events. This characteristic makes DPRA particularly relevant for cyber-security risk assessment Prescott (2025), as it enables the analysis of attack scenarios that may be difficult or impossible to identify in advance or using static methods.

In this paper, we analyse a cyber-attack scenario in the IT domain using a DPRA toolbox. The considered cyber threat can be summarized as follows: a user receives emails containing malicious links and may inadvertently click on them, while regularly accessing a high-performance computing system, which is the ultimate target of the attack. The system behaviour is modelled using the statecharts visual modelling language.

The remainder of this paper is structured as follows. First, the general principles of statechart modelling for risk assessment are introduced. Next, the detailed system model is presented. Finally, simulation results, risk importance measures, and the main insights derived from the analysis are discussed.

2. Statecharts as a formalism for dynamic reliability modeling

Statecharts were initially introduced to enable the modelling of complex reactive systems Harel (1987). More recently, they have been adopted for dynamic reliability modelling by extending the formalism with stochastic features Rychkov et al. (2023a). A derivative of a commercial low-code development platform, *itemisCREATE*, together with its dynamic reliability plugin Rychkov et al. (2025), allows the creation, simulation, and quantification of statechart models representing the system under consideration.

Figure 1 illustrates a statechart model of a system composed of two redundant components arranged in a standby configuration Rychkov et al. (2023b). Each component can fail while running and can be repaired. The system is simulated over the time interval $t \in [0, t = missionTime]$. At $t = 0$, *Component1* is requested to start. *Component2* operates in cold standby mode and is requested to start only when the first component is no longer operational. The startup of *Component2* requires a fixed duration of $startTime$ seconds.

The safety function is considered fulfilled if at least one of the two components is in an operational state. If the safety function is not fulfilled for more than $graceTime$ seconds, the system transitions to an *EndState*, representing the overall system failure.

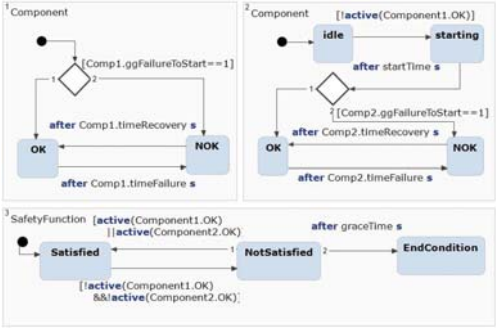

Fig. 1. A statechart model of a system composed of two redundant components with a standby redundancy.

The model shown in Figure 1 includes both fixed and random variables. The EDF-developed reliability plugin for *itemisCREATE* enables the specification of these parameters directly within the model. Figure 2 illustrates the declaration of the variables used in the model.

Fig. 2. Variable declaration of the dynamic reliability statechart model.

Component failure and recovery times, denoted respectively as $timeFailure$ and $timeRecovery$,

are sampled from exponential distributions with parameters $lambdaComponent$ and $muComponent$. A failure-on-demand event, denoted as $ggFailureToStart$, is sampled from a binomial distribution parameterized by $gammaComponent$. The parameters $graceTime$ and $startTime$ are deterministic and remain fixed for a given simulation.

To quantify a risk indicator associated with the statechart model, a Monte Carlo simulation is performed using a dedicated configuration file. Figure 3 presents an example of such a configuration, which specifies parameters related to the simulation setup (e.g., the number of Monte Carlo samples and random number generator initialization), as well as reliability parameters and fixed-value parameters used during the simulation. As an outcome of the simulation, the time evolution of all individual sequences that reach the *EndState* of the safety function region is stored and can be subsequently analysed Picoco and Rychkov (2024).

```
{
  "numberOfSimulations": 1000, //number of MonteCarlo Simulations
  "missionTimeSimulated": 86400, //seconds
  "endState": "SafetyFunction_EndCondition", //end state
  "seed": 1234, //Random number generator initialization
  "root": {
      "lambdaComponent": 1e-6, // 1/s
      "gammaComponent": 1e-10, // per start
      "muComponent": 1e-5, // per second
      "graceTime": 20, //seconds
      "startTime": 10 //seconds
  }
}
```

Fig. 3. Example of a Monte Carlo simulation configuration file.

In the next section, the statechart model of the system under a cyber-attack scenario is presented.

3. Modeling a cyber atack using statecharts

We model the following cyber-attack scenario: a user receives emails containing malicious links and may inadvertently click on such links while regularly accessing a high-performance computing (HPC) system, which represents the ultimate target of the attack. The model incorporates multiple protection mechanisms, including antivirus software, system updates, and firewalls. Risk importance measures are computed to gain insights into the attack dynamics and to assess the effectiveness of different security barriers over an attack duration of one week.

The description of the attack combines behavioural aspects (e.g. receiving emails, user clicking behaviour) and structural aspects (e.g. the user's personal computer and the HPC system). One of the main modelling challenges is to represent both the behavioral features and the system structure coherently within a single statechart model Marron et al. (2018).

Figures 4–7 present the general overview of the statechart model, which is composed of several interacting subsystems.

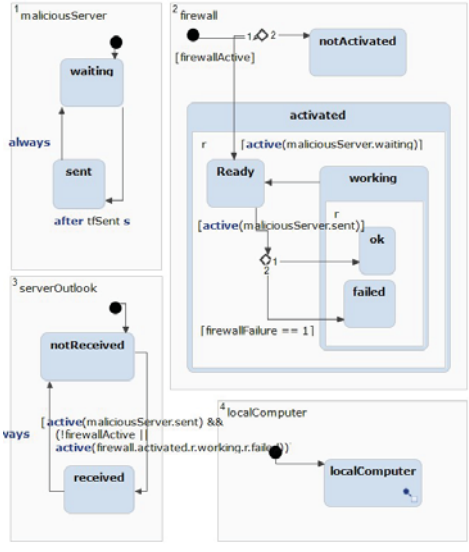

Fig. 4. Overview of the system under cyber attack, part 1: malicious server, firewall, Outlook server, and local computer.

In the following subsections, the different elements of the model are described in more detail.

3.1. *Malicious server*

The malicious server sends emails containing malicious links at time intervals characterized by a random variable $tfSent$. The email sending process is modelled using an exponential distribution with an average frequency of four emails

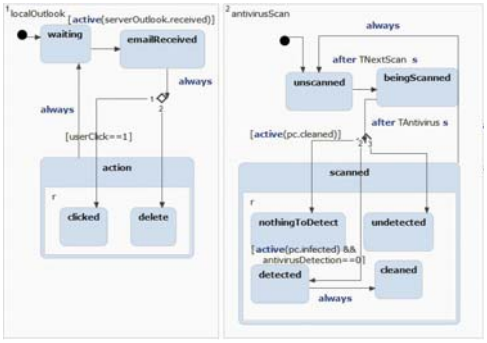

Fig. 5. Local computer model, part 1.

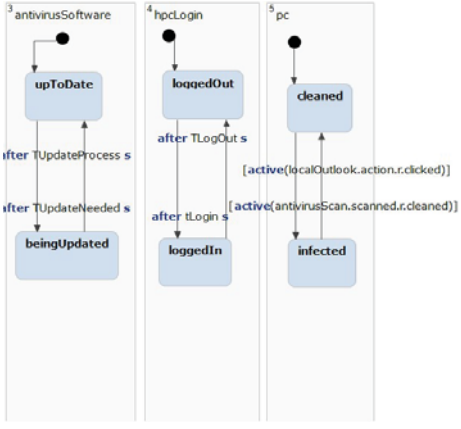

Fig. 6. Local computer model, part 2.

per day, corresponding to a rate parameter of $lambdaSend = 4.63 \times 10^{-5}\, s^{-1}$.

3.2. *Firewall*

A firewall protects both the local Outlook mail server and the user's computer from malicious emails. When activated, the firewall has a probability of failure $pFireWallFailure = 1 \times 10^{-3}$ per malicious email, allowing such an email to pass through. If the firewall is deactivated due to maintenance or malfunction, it allows 100% of the emails to pass. The initial state of the firewall (activated or deactivated) is fixed at the beginning of the Monte Carlo simulation and remains unchanged for all simulated sequences.

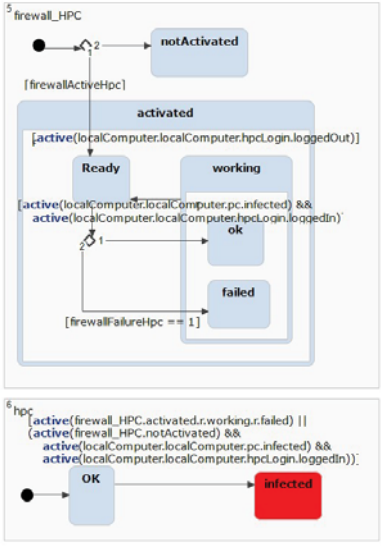

Fig. 7. Overview of the system under cyber attack, part 2: HPC firewall and HPC system.

3.3. *Outlook mailing server*

If the firewall allows a malicious email to pass, the Outlook mail server receives the message and makes it available for dispatch to the local computer.

3.4. *Local computer*

The structure of the local computer model is shown in Figures 5 and 6.

3.4.1. *Local Outlook client*

Once the Outlook server receives an email containing a malicious link, the local Outlook client is assumed to be automatically synchronized. The user can then either open the email and click on the embedded link or delete the email without interacting with it.

We assume that the probability of clicking on a malicious link depends on the user profile. The user profile is selected with probability $pProfile = 0.5$ between an expert user, who has presumably followed cyber-security training and clicks on the link with probability $pUserExpert = 0.1$, and a beginner user, who clicks with probability $pUserBeginner = 0.4$.

3.4.2. *Antivirus program: scanning and updates*

An antivirus program is installed on the local computer. Periodic scans are performed every 24 hours ($TNextScan = 86400\,s$), and each scan lasts 30 minutes ($TAntivirus = 1200\,s$). The antivirus fails to detect an existing infection with probability $pAntivirus = 0.2$.

The antivirus software is updated every 72 hours ($TUpdateNeeded = 259200\,s$), with each update lasting 10 minutes ($TUpdateProcess = 600\,s$). If the antivirus fails to detect the malware after the user clicks on the malicious link, it is assumed that detection is not possible until the next update process is completed.

3.4.3. *Local computer state*

If the user clicks on a malicious link, the local computer becomes infected. If the antivirus successfully detects the malware, the computer returns to an uninfected (clean) state.

3.4.4. *Remote login to the HPC*

The user logs into the HPC system at random intervals ranging from 5 to 7 hours ($minLoginDist = 18000\,s$, $maxLoginDist = 25200\,s$) and remains logged in for a duration of 1 hour ($TLogOut = 3600\,s$).

3.5. *HPC firewall*

The HPC system is protected by a firewall operating according to the same principles described in Section 3.2. If activated, the firewall fails to block malware with probability $pFirewallFailureHpc = 1 \times 10^{-3}$ per malware transmission attempt. If deactivated, all malware passes through unfiltered.

3.6. *HPC*

The ultimate objective of the attacker is to compromise the HPC system. Infection occurs if the local computer is infected at the moment of user login and the HPC firewall is ineffective, either because it is deactivated or because it fails to filter the malware.

3.7. *Input data discussion*

Table 1 summarizes the model parameters introduced above. The numerical values were defined in consultation with cyber-security experts and are intended to be representative rather than specific to any particular technology or operational system.

Table 1. Model parameters

Parameter	**Value**
lambdaSend	4.63E-5 s^{-1}
minLoginDist	18000 s
maxLoginDist	25200 s
pProfile	0.5
pUserBeginner	0.4
pUserExpert	0.1
pAntivirus	0.2
pFirewallFailure	1E-3
pFirewallFailureHpc	1E-3
TAntivirus	1200 s
TLogOut	3600 s
TNextScan	86400 s
TUpdateNeeded	259200 s
TUpdateProcess	600 s

4. Results

The model described in Section 3 was quantified to estimate the risk of a successful cyber attack, here defined as the probability of *HPC infection* over a one-week mission time. The probability of HPC infection is 2.7×10^{-5} for the countermeasure performance parameters in Table 1.

In this paper we do not intend to discuss the absolute value of the risk associated wit this scenario, instead, we focus on insights derived from the analysis.

Probabilistic Risk Assessment (PRA) typically delivers two principal categories of insights: (i) identification of *failure (or attack) scenarios*, and (ii) ranking of contributors via *importance measures*. Dynamic reliability methods (DPRA) are particularly suited to the first category because they explore system behaviour through simulation, generating relevant risk scenarios without requiring an *a priori* enumeration. This feature is valuable for complex systems, where specifying

all attack paths beforehand is challenging.

In the present study, the dominant attack scenario can be summarized as follows: the perimeter firewall allows a malicious email to reach the Outlook server and the user's local computer; the user clicks the malicious link, infecting the local computer; the antivirus either fails to detect the infection or does not scan before the next HPC login; finally, the HPC becomes infected if, at login, the HPC firewall is ineffective (either deactivated or failing to filter the malware).

To assess the relative effectiveness of security barriers, we adopt importance measures from traditional PRA Vesely et al. (1983). We compute the Risk Increase Factor (RIF, also known as RAW) and the Risk Reduction Worth (RRW) for the following parameters:

- **User training / profile mix:** we assume the user population can shift toward experts via training. For RIF, we set the population to "all beginners"; for RRW, we set all population to "all experts".
- **Antivirus update frequency:** to assess the value of more frequent signature updates. For RIF, updates are disabled during the mission time; for RRW, updates occur every 15 minutes.
- **Antivirus detection reliability:** for RIF, we disable detection ($pAntivirus = 1$); for RRW, we assume the perfect detection ($pAntivirus = 0$).
- **Firewall reliability (perimeter and HPC):** for RIF, we disable the respective firewall ($pFirewall = 1$); for RRW, we assume a perfect firewall ($pFirewall = 0$).

Table 2 presents the computed importance measures. We report RAW (RIF) and the inverse of RRW (denoted RRW^{-1}), which represents the fraction of residual risk under a perfect barrier (values closer to 0 indicate larger potential risk reduction).

These results indicate that performance of both firewalls dominantes the risk profile: large RAW values ($\sim 10^2$–10^3) imply that disabling either firewall increases the risk by orders of magnitude, and $\text{RRW}^{-1} \approx 0$ suggests near-elimination of the cyber-risk under perfect filtering. The antivirus exhibits a moderate RAW (≈ 3.2), indicating that maintaining its current detection performance is important to avoid risk escalation.

Table 2. Importance measures for different model parameters

Parameter	RAW (RIF)	RRW^{-1}
pProfile	1.5	0.3
antivirusUpdateF	1.1	0.9
pAntivirus	3.2	0.6
pFirewall1	406.0	0.0
pFirewall2	347.3	0.0

User training (profile mix) yields a comparatively stronger *risk reduction* potential ($\text{RRW}^{-1} = 0.3$) than *risk increase* potential ($\text{RAW} = 1.5$), suggesting that shifting the population toward expert behaviour is an efficient lever for reducing risk. By contrast, increasing the update frequency alone (with current detection efficacy) has limited marginal benefit over a one-week horizon ($\text{RAW} = 1.1$, $\text{RRW}^{-1} = 0.9$).

The application of *dynamic* importance measures (e.g., time-dependent, path-conditional metrics) to cyber-security models is a promising direction and will be addressed in future work Jankovsky et al. (2016); Mandelli et al. (2019).

5. Conclusions

In this paper we present a behavior model of a cyber-attack using statecharts. We develop the model using a reliability extension (a plugin) for the itemisCREATE "low code" tool. We simulate the model to quantify the risk of the successful cyber attack as well as the importance measures for different protective measures. This work demonstrates that dynamic reliability toolbox can be applied for cyber security analysis.

References

Blaszkiewicz, M. et al. (2024). Availsim4 – an open-source framework for availability and reliability simulations. In *Advances in Reliability, Safety and Security. ESREL Contributions*.

Harel, D. (1987). Statecharts: A visual formalism

for complex systems. *Science of Computer Programming 8*(3), 231–274.

IRSN (2024). Avis irsn n° 2024-00049: Établissement orano recyclage de la hague — inb n° 116 (up3-a) et inb n° 117 (up2-800). études probabilistes de sûreté en réponse aux engagements n° 9 et n° 17 pris dans le cadre des réunions du groupe permanent d'experts pour les laboratoires et les usines relatives au réexamen périodique de l'usine up2-800. Technical report, ASNR / IRSN. Accedé 12/2025.

Jankovsky, Z. K., M. R. Denman, and T. Aldemir (2016). Dynamic importance measures in the adapt framework. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Lallie, H. S., K. Debattista, and J. Bal (2020). A review of attack graph and attack tree visual syntax in cyber security. *Computer Science Review 35*, 100219.

Mandelli, D., Z. Ma, C. Parisi, A. Alfonsi, and C. Smith (2019). Measuring risk-importance in a dynamic pra framework. *Annals of Nuclear Energy 128*, 160–170.

Marron, A., Y. Hacohen, D. Harel, A. Mülder, and A. Terfloth (2018). Embedding scenario-based modeling in statecharts. In *MoDELS (Workshops)*, pp. 443–452.

Moulié, F. et al. (2022, October). Analyses probabilistes de sûreté innovantes avec modélisations dynamiques. In *Congrès Lambda Mu 23: Innovations et maîtrise des risques pour un avenir durable*, Paris-Saclay, France. IMDR. hal-03966660.

Picoco, C. and V. Rychkov (2024). Bridging the gap between static and dynamic probabilistic assessment: a step forward. In *PSAM 17 / ASRAM 2024*, Sendai, Japan.

Picoco, C., V. Rychkov, and T. Aldemir (2020). A framework for verifying dynamic probabilistic risk assessment models. *Reliability Engineering & System Safety 203*, 107099.

Prescott, S. e. a. (2025). Performing numerical analysis of cybersecurity options using dynamic risk analysis tool emrald. In *ANS PSA 2025*.

Rychkov, V. et al. (2023a). Statecharts as a dynamic method for risk assessment. In *Topical Meeting on Probabilistic Safety Assessment (PSA 2023)*, Knoxville, USA.

Rychkov, V. et al. (2023b). Statecharts as a dynamic method for risk assessment. In *Topical Meeting on Probabilistic Safety Assessment (PSA 2023)*, Knoxville, USA. Variante de la référence PSA 2023.

Rychkov, V. and C. Picoco (2022). Model based software engineering techniques for dynamic reliability assessment. In *ESREL 2022*, Dublin, Ireland.

Rychkov, V., C. Picoco, and A. Terfloth (2025). Statechart based dynamic pra tool. In *ANS PSA 2025*.

Vasilyev, A., J. Andrews, S. J. Dunnett, and L. M. Jackson (2021). Dynamic reliability assessment of pem fuel cell systems. *Reliability Engineering & System Safety 210*, 107539.

Vesely, W., T. Davis, R. Denning, N. Saltos, et al. (1983). Measures of risk importance and their applications. Technical report, Battelle Columbus Labs., OH (USA).