CITY
UNIVERSITY OF LONDON
EST 1894

Academic excellence for business and the professions

Models of reliability of fault-tolerant software under cyber-attacks

Peter Popov,
Centre for Software Reliability

23 March 2017
Urbana-Champaigh

Software Reliability in Adverse Environment

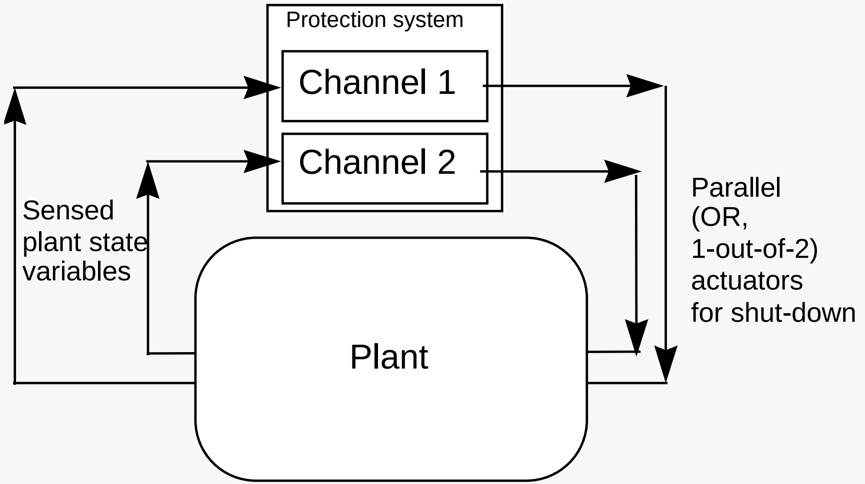
- A significant effort on “co-engineering” for safety – reliability-performance has been spent
  - People talk about trade-offs, interdependencies between multiple non-functional properties, but models which make the dependence explicit do not seem to exist
  - Safety standards (e.g. ISO 61508, the main safety standard) recommend that “all risk including from malicious activities” be taken into account when engineering system safety.
  - The industrial practice is not ideal – different “silos” make it difficult to co-engineer for safety and security since different departments deal with different concerns.

Motivation

- In this work I make an attempt to model the impact of successful cyber-attacks on software reliability *explicitly*.
- The work targets primarily *industrial control system*,
  - These must work reliably, availability is a paramount concern
  - privacy and confidentiality **are typically not** a primary concern
    - There may be exceptions – smart metering infrastructure is an example.
- Having reviewed a large number of reports of attacks on industrial control system convinced me that the model put forward *is plausible*.
  - Full validation of the assumptions is yet to be done.

System model

- 1-out-of-2 “on demand” software
  - Popular in safety critical applications, e.g.
    - Protection systems
    - Automotive industry (ASIL-D according to ISO 26262)
    - Etc.



System model (2)

- Channels can be subjected to cyber attacks (i.e. ***malicious demands***)
- The view taken by many is: “once a malicious demand succeeds, the game is over: the adversary can do whatever they please”
  - The consequences of successful malicious demands are not modelled in detail and “the worst” consequences are assumed.
- In this work I take a *different view*:
  - Successful demands merely *increase the probability of failure* on “normal” (i.e. non-malicious) demands.
    - Immediate failure after a successful attack is merely a special case of *extreme reliability decay*: the probability of failure on demand increases to 1 (deterministic failure).

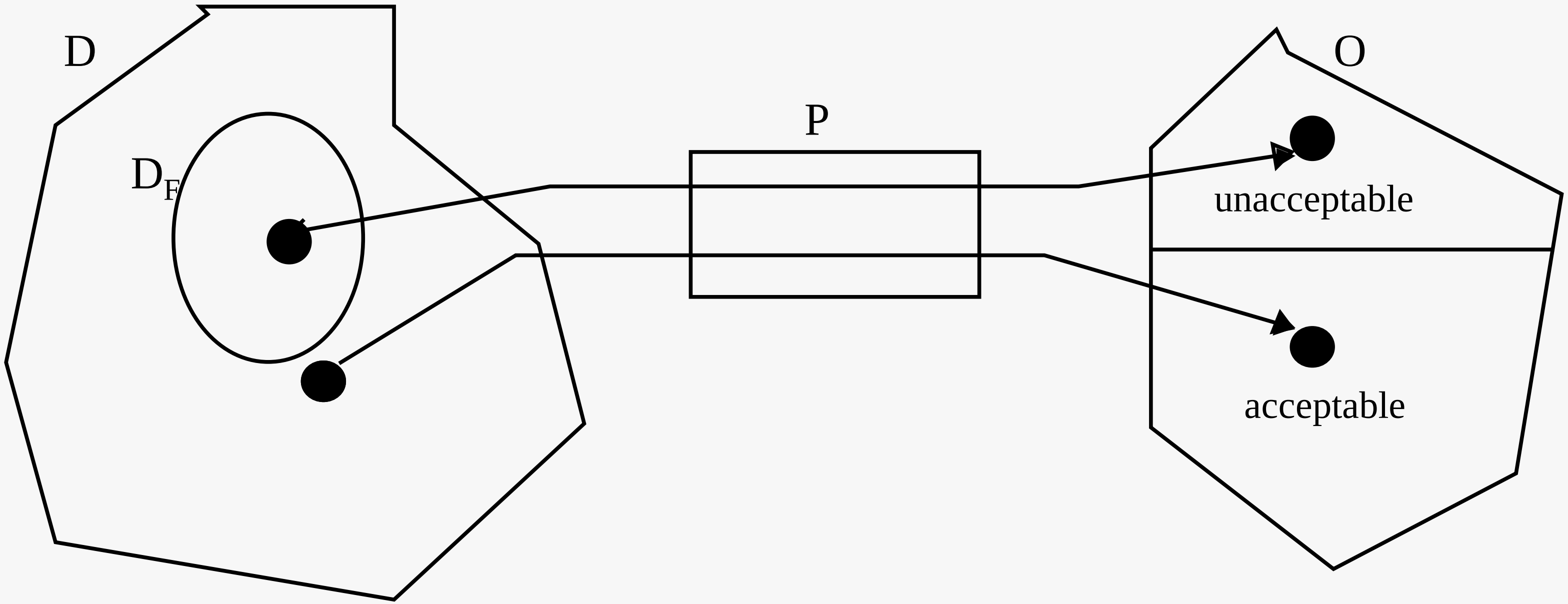
System model (3)

- Channels are “diverse” (using software design diversity)
  - Design diversity has been studied very extensively at City
  - Diversity ***does not guarantee failure independence***
- Demands are *independently selected from the demand space* and processed by each channel independently

- The system fails when both channels fail simultaneously on the same demand
  - The probability of system failure on demand X (*pfd*) is:

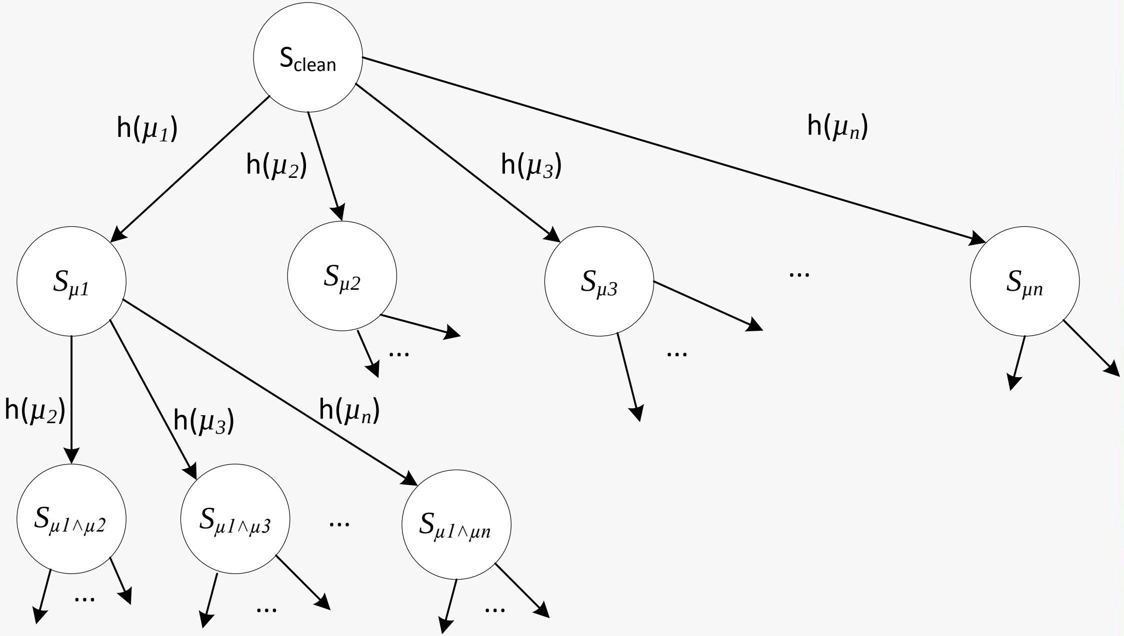
$$P(\pi_A, \pi_B\, fail\ on\ X) = pfd_A pfd_B + cov_Q\big(\omega_A(X, \pi_A), \omega_B(X, \pi_B)\big)$$

Probability of failure on demand



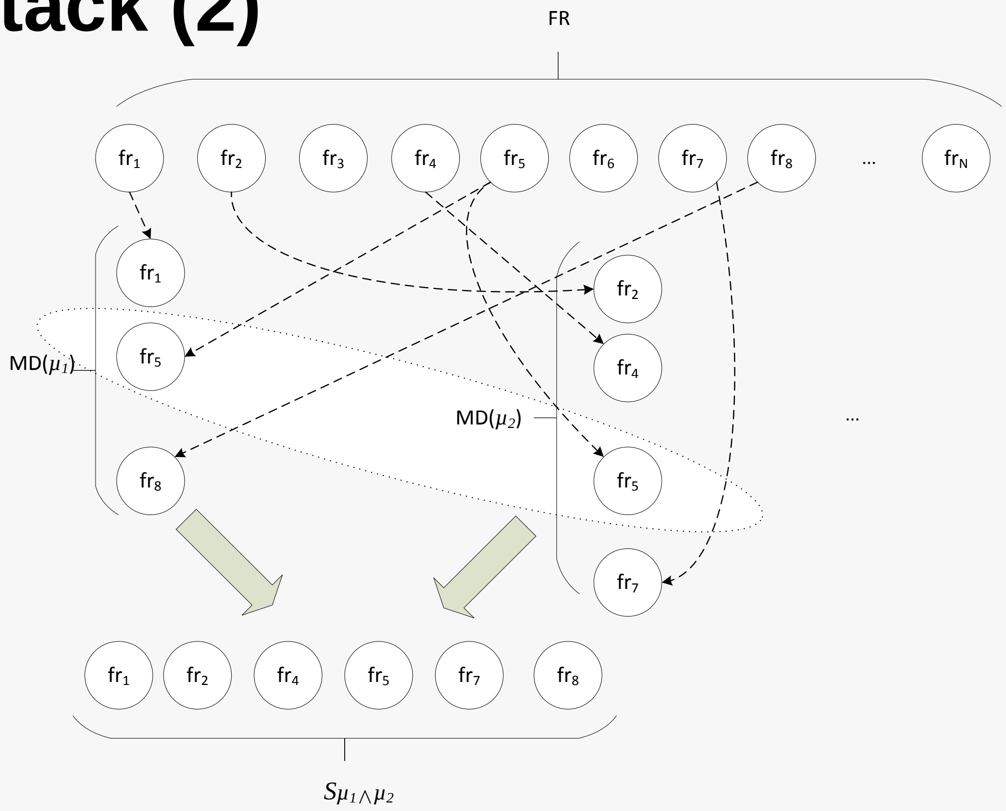
System under-attack

- Malicious demands (MD), $\{\mu_1, \mu_2, \ldots, \mu_n\}$ can be applied to each of the channels.
- MD are either successful or blocked (e.g. by an intrusion protection system.
- Demands are *serialized* (i.e. at most one demand is applied at a time)



System under-attack (2)

- Each successful malicious demand may introduce new “***failure regions***” on the demand space of the attacked channel.
- If more than 1 malicious demand succeeds, the *union* of the respective failure regions is added to the demand space of the affected channel.



System under-attack (3)

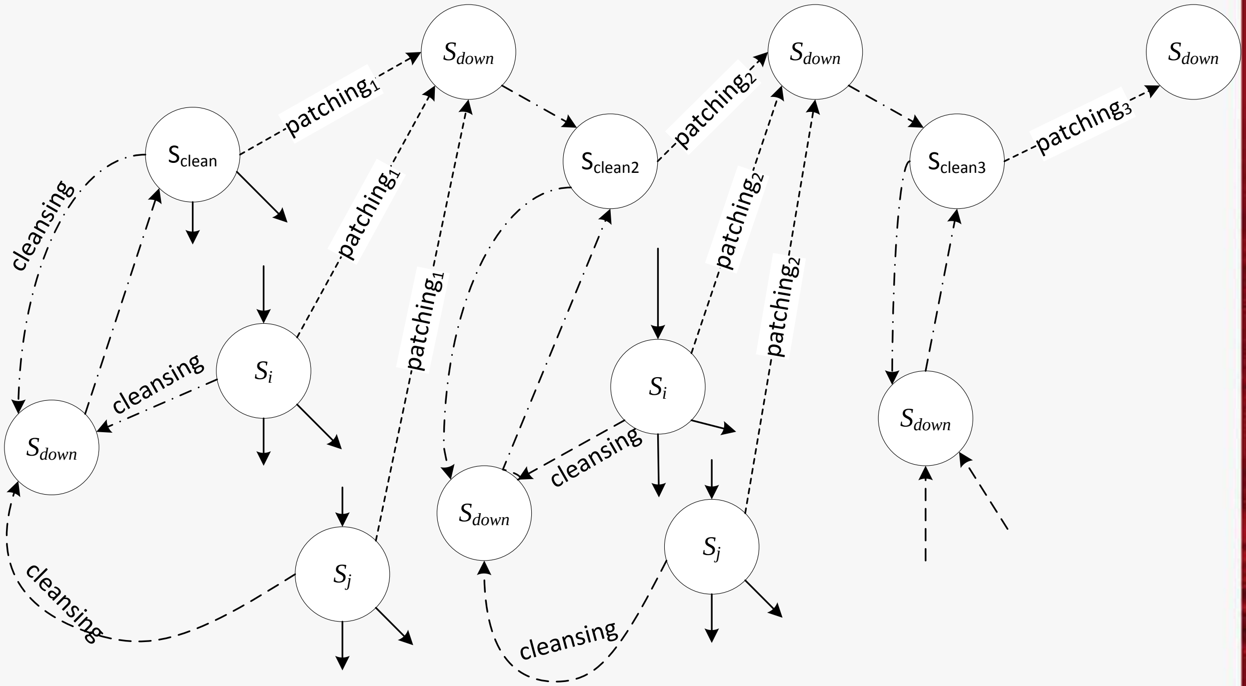
- If the failure regions introduced by attacks on both channels *overlap* (i.e. there are demands that belong to a failure region in both channels), then system *pfd* increases by the size of the overlap.
- Malicious demands can be:
  - "Independently" applied to channels (i.e. the adversary is unaware that they are dealing with a 1-out-of-2 architecture);
    - Even with independent demands system *pfd* can go up – a random overlap of failure regions caused by malicious demands applied to both channels;
  - "Synchronized", when the adversary is aware that they attack a two channel system
    - In the extreme case, the same set of new failure regions will be added to the demand space of ***both channels***, possibly with some delay (we assume that simultaneous attacks on both channels are impossible).

System maintenance

- Channels can be ***periodically*** maintained.
- We consider two types of maintenance
  - “Cleansing”
    - similar to rejuvenation, but more than it – restoring the installation ***from a clean copy***; rejuvenation is typically just a reboot.
    - Cleansing eliminates all failure regions introduced by successful malicious demands on the demand space of the particular channel.
  - Patching
    - Similar to cleansing, but also:
      - Reduces the initial channel *pfd* (possibly the system *pfd,* too) – since ***some bugs may have been fixed***
      - May also reduce the probability of success of some malicious demands (possibly to 0) by ***eliminating exploitable vulnerabilities***.

System maintenance (2)

- System modelled as a stochastic ***state machine*** with an evolving “channel state” (the state is the set of failure regions added and removed by malicious demands and maintenance):



System maintenance (3)

- Maintenance with a 2-channel software must be managed to avoid maintaining more than one channel at a time:
  - *mutex* of channel maintenance
  - During maintenance, the system is reduced to a single channel
    - Hence the system *pfd* becomes equal to the *pfd* of the remaining operational channel

- Maintenance regimes ***can be combined*** in 4 different scenarios:
  - No maintenance (this is the reality!)
  - Cleansing only
  - Patching only
  - Cleansing and patching

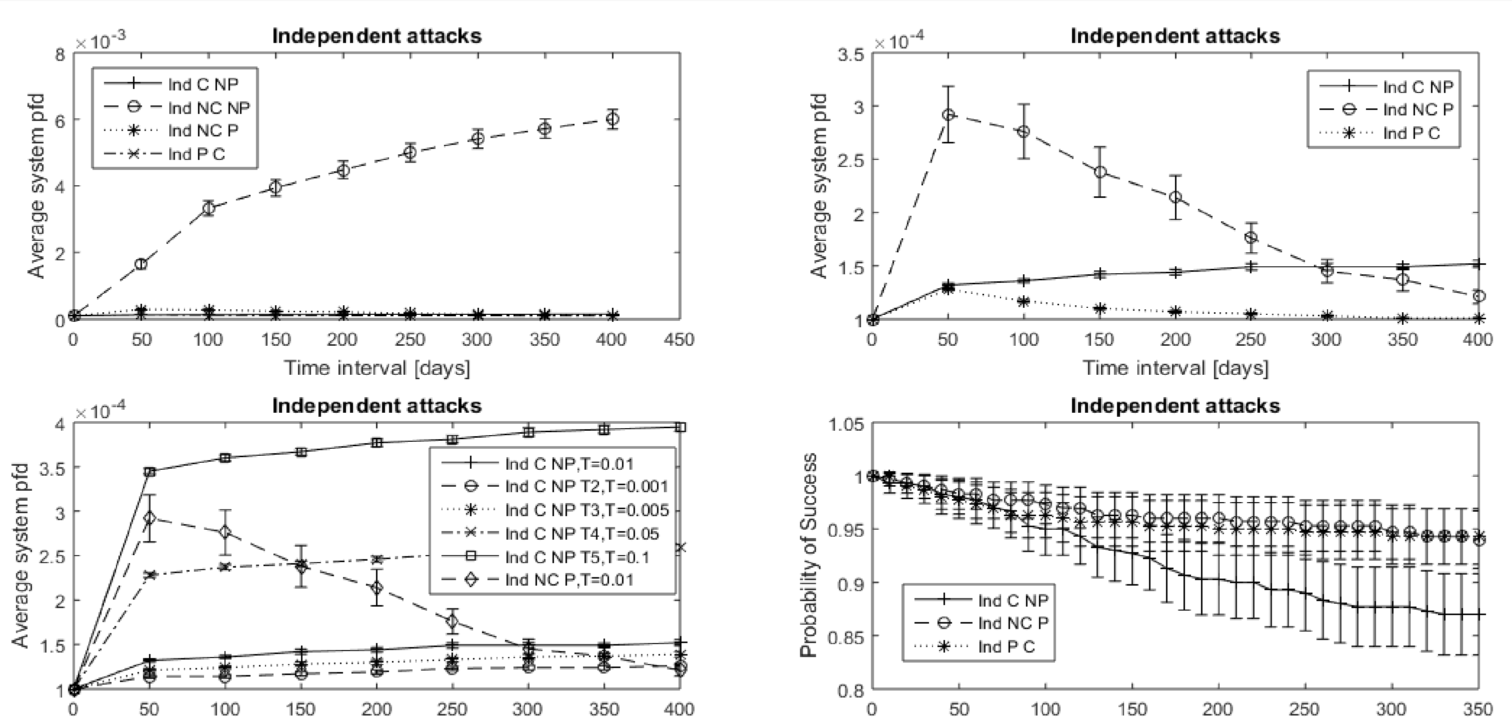
Studies

- A probabilistic model was built to study the effects of attacks on the channels of a 1-out –of-2 system
  - The SAN formalism was used with some custom C code added.
  - Parameterization selected to allow for:
    - Different maintenance regimes
    - Different frequencies of maintenance
    - Different attack types (independent vs. synchronized)
  - Looked at different measures of interest among them:
    - Mean Probability of failure over a time intervals
    - Mean Time to system failure (i.e. until both channels fail on the same demand)
      - In the studies I fixed the frequency of the normal demands.
  - The model is solved via Monte Carlo simulation
    - 350 days of operation
    - Measures also calculated over 8 sub-intervals – to capture trends.

Model parameters

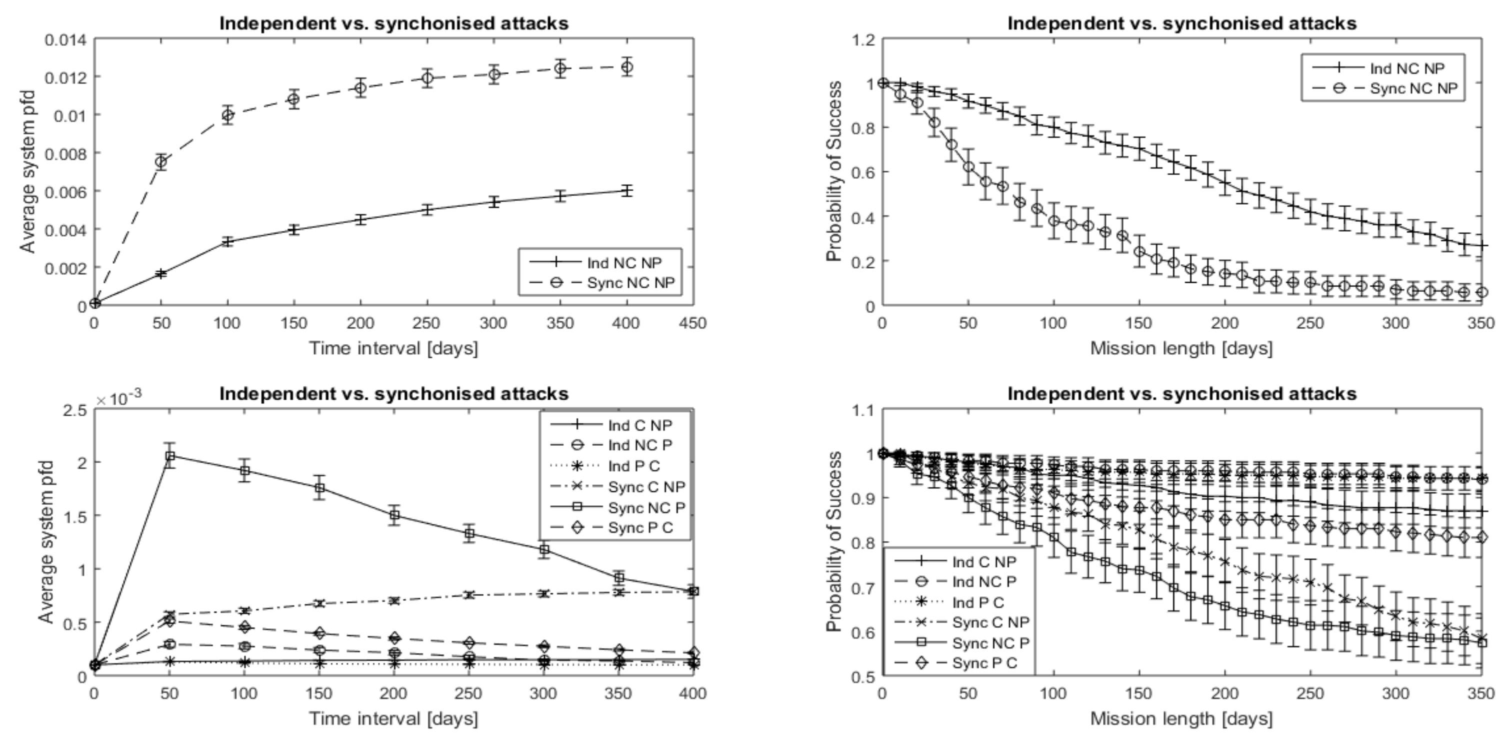
Name	Description	Value
FR_size [day^{-1}]	Failure region sizes (uniformly distributed)	1.00E-04 – 2.00E-4
attackRateIncrease [day^{-1}]	Attack rate *increase* over time Changes to rates are applied in the model upon channel patching.	0.001
attackSProb_reduction	Coefficient of reduction of the malicious demand probability of success.	0.95
ch1_pfd	Channel 1 *pfd* (no malicious demands)	0.002
ch2_ pfd	Channel 2 *pfd* (no malicious demands)	0.002
cleansing_interval [day]	Intervals between cleansings of channels (if cleansing is enabled in the particular study)	1
common_pfd	System *pfd* (before any channel is compromised).	1.00E-04
delay_sync_attack [day]	Delay between channel attacks in case channels are attacked synchronously.	0.05
demand_rates [day^{-1}]	Malicious demands rates (exponential distributions)	0.01-0.07
demand_types	Number of demands in channel 1 and channel 2 (could differ).	10
m_demand_max_FRs	Maximum failure regions per malicious demand	10
normal_demands_rate [day]	Interval between normal demands on 2-channel system.	1
patchingInterval [day]	Interval between patches (exponential distribution)	0.15
patch_ch_pfd_reduction	Channel *pfd* reduction coefficient after patching.	0.9
patch_CC_pfd_reduction	System *pfd* reduction coefficient after patching a channel.	0.95
upgrade_duration [day]	Maintenance duration (fixed interval)	0.01

Results



Independent attacks
×10^-3
Average system pfd
Time interval [days]
Ind C NP
Ind NC NP
Ind NC P
Ind P C
Independent attacks
×10^-4
Average system pfd
Time interval [days]
Ind C NP
Ind NC P
Ind P C
Independent attacks
×10^-4
Average system pfd
Ind C NP,T=0.01
Ind C NP T2,T=0.001
Ind C NP T3,T=0.005
Ind C NP T4,T=0.05
Ind C NP T5,T=0.1
Ind NC P,T=0.01
Independent attacks
Probability of Success
Ind C NP
Ind NC P
Ind P C

Results (2)


Independent vs. synchonised attacks
Average system pfd
Time interval [days]
Ind NC NP
Sync NC NP
Independent vs. synchonised attacks
Probability of Success
Mission length [days]
Ind NC NP
Sync NC NP
Independent vs. synchonised attacks
× 10^-3
Average system pfd
Time interval [days]
Ind C NP
Ind NC P
Ind P C
Sync C NP
Sync NC P
Sync P C
Independent vs. synchonised attacks
Probability of Success
Mission length [days]
Ind C NP
Ind NC P
Ind P C
Sync C NP
Sync NC P
Sync P C

Summary of the observations

- The effect of attacks on fault-tolerant software depends very significantly on the *model of attacks*
  - *not surprising, but some insight obtained with the model*
- Assessment assuming independent attacks (including those by red-teams/pen-testing) may give ***dangerously optimistic conclusion*** about how good the 2-channel system is.
  - The Target environment must be carefully considered and if synchronized attacks cannot be ruled out – apply synchronized attacks in the assessment.

Summary (2)

- Malicious demands may lead to significant variation of system reliability over time:
  - All estimates of the probability of successful mission are much worse than what they should have been with the average system *pfd*. Variation of system *pfd is* clearly harmful.
  - Established techniques for reliability assessment based on assuming ***constant pdf*** throughout the entire mission (say a year, etc.) need to account for this variability.
- The model is available at:

http://openaccess.city.ac.uk/16700/.

Implications for other replication systems

- Any replication scheme is guaranteed to works correctly under a number of assumptions:
  - 1-out-of-2: at least one of the channels must be correct
    - Taking down a channel for maintenance makes the 2-channel system a mere 1-channel system.
  - A number of *intrusion-tolerant architectures* are based on a Byzantine agreement protocol + cleansing in different flavors (“proactive recovery”, “proactive obfuscation”, etc.). These rely on a number of assumptions:
    - Replication is guaranteed to work correctly only if the number of compromised replicas is smaller than a given threshold
    - Can this assumption be enforced? How?
      - If not, can it be violated. How likely is such violation of the assumptions? The presented model allows for some exploratory analysis to be undertaken.

Future work

- The presented model is a “conceptual one”.
  - Can be used to ***get insight***, but is probably useless for reliability prediction.
- Can the model be improved and made suitable for predictions? Maybe.
- More seriously (work in progress)
  - One needs to validate the inherent assumption built in the model construction that attacks decrease reliability. This needs to be done more thoroughly than has been done so far.
  - Data needed for validation may pose serious difficulties:
    - we have several stochastic processes, and some are not fully observable.

Questions

- Thank you!

City, University of London
Northampton Square
London
EC1V 0HB
United Kingdom

T: +44 (0)20 7040 8963
E: p.t.popov@city.ac.uk
www.city.ac.uk/people/academics/peter-popov