Cut-The-Rope: A Game of Stealthy Intrusion

S. Rass[1] (presenter), Sandra König[2], Emmanouil Panaousis[3]

[1] Universität Klagenfurt, Austria, [2] Austrian Institute of Technology, Austria, [3] University of Surrey, UK

stefan.rass@aau.at, sandra.koenig@ait.ac.at, e.panaousis@gmail.com

30.10.19

Contents

Motivation 1

Characteristics of contemporary cyber attacks:

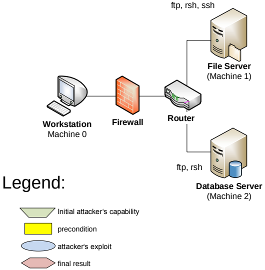
- stealthy
- no noticeable start (often by a harmlessly looking email, malicious USB device, ...)
- adapted to the victim system (defender's "usual" moves, ...)

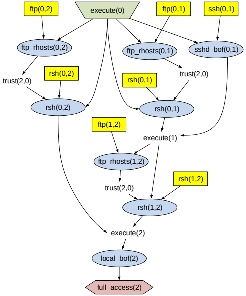
$\rightarrow$ Conventional game theoretic models are difficult to apply for practitioners

Why conventional game theoretic models are difficult to apply for practitioners:

- the game is not round based
- it has no defined start, but a defined finish (the loss of the target asset)
- defender cannot cope with an arbitrary lot of losses $\rightarrow$ average loss often makes no sense to optimize, since losses normally don't average, but accumulate!
- accurate models often come with too many parameters (that are hard to understand and even harder to instantiate)

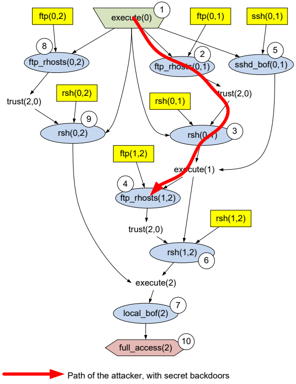
- our challenge is to defend an infrastructure from intruders, seeking to access a database server (machine 2) with sensitive personal information on it (e.g., customer data).
- Such attacks usually start from some workstation (here machine 0) (initial access established by social engineering)





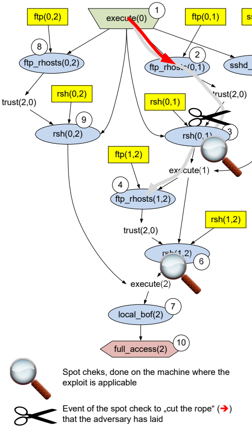
The playground is an attack graph, on which the attacker (inspectee) tries to hiddenly get to the target node, here ⑩, from a starting node ①, while the defender (inspector) tries to prevent that.

- Along the way, the attacker installs backdoors for an easy comeback to learn how to get over to the next node (closer to the target).



The playground is an attack graph, on which the attacker (inspectee) tries to hiddenly get to the target node, here ⑩, from a starting node ①, while the defender (inspector) tries to prevent that.

- Along the way, the attacker installs backdoors for an easy comeback to learn how to get over to the next node (closer to the target).
- The defender spot-checks the system, and – even unknowingly – cleans a machine from a backdoor, e.g., by patching, updating, changing passwords, etc.



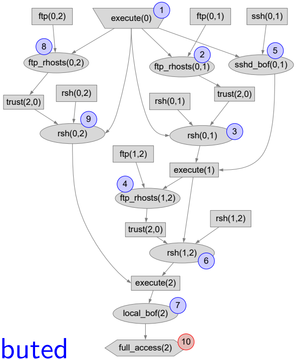
The Inspector/Defender's Uncertainty 1

A central characteristic of APTs is their stealthiness, so the defender has. . .

- no idea from where the attacker may start, or how far it has already made it into the system.
- no means of knowing when the game originally started (even if it started already), or for how many rounds it has been played.
- The only noticeable event is the attacker having reached the target → by then, the inspector is "effectively dead", and the game is lost.
- typically fixed schedules of becoming active (during daily working hours), while the attacker can move at any point in time (the game is round-based for the defender, while it is in continuous time for the attacker → this "asynchronicity" is usually not found in most game theoretic models.

The Inspector/Defender's Uncertainty 2

- Defender never "sees" the attacker (or knows its location)
- We consider one adversary type per possible location (excluding the target ⑩), each of which induces its own strategies (paths towards ⑩ and exploits thereon)
- Bayesian game to capture the invisibility

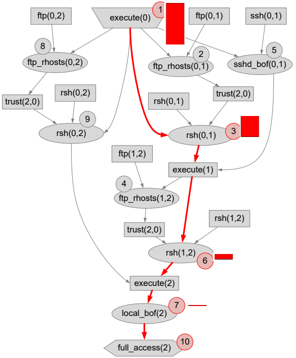


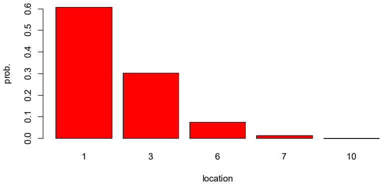
→ payoff = distance to v_0, Poisson distributed

→ stochastically ordered (keep the attacker away from v_0, optimizing an average distance is meaningless here).

Defining the Payoffs 1

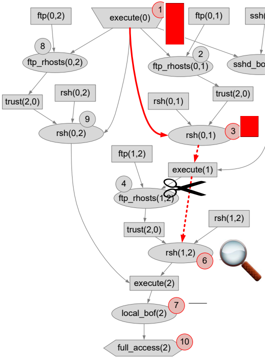
- For example, taking attack path #5, the attacker will move a random number of N steps further down on the red route $\rightarrow$
- The payoff is thus a random variable, putting a probability mass on each upcoming location according to the Poisson distribution.
- This is a payoff for a type $\theta = 1$ adversary, taking its pure strategy (attack path #5), conditional on the defender not intercepting this path yet.

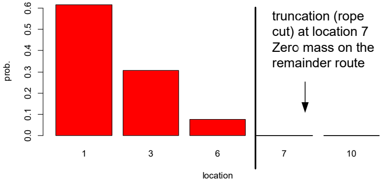




Defining the Payoffs 2

- If the defender, however, intercepts the route, say by spot-checking at location ⑥, which is one of the defender's pure strategies, then the path gets cut,
- and the payoff distribution becomes truncated ↘





Game Model

- Game model: "simple" matrix game
  - defender's actions: all nodes (in the attack graph) that admit spot checking
  - adversary's actions: all attack paths
- payoffs: determined by truncating distributions
- optimization of tail mass = probability to hit target asset v_0
- equilibrium: perfect Bayesian (here, equivalent to a multi-criteria security strategy, by fictitious play)
- easy to implement (taking $\approx$ 30 lines of code)
  - implemented in `R`, with the `HyRiM` package
  - for the example, take $\lambda = 2$, i.e., an average of two penetration steps being accomplished per time unit.
  - The full code is available for free download from `https://www.syssec.at/de/downloads/papers`

Summary

CUT-THE-ROPE features:

- asynchronous movement, discrete time player vs. continuous time opponent
- strong asymmetry: only one out of two players knows when the game starts
- ease of implementation (takes only a few lines of code to run)
- ease of generalization, like:
  - randomized defense actions
  - probabilistic success (of the defender and/or attacker)
  - multiple target assets

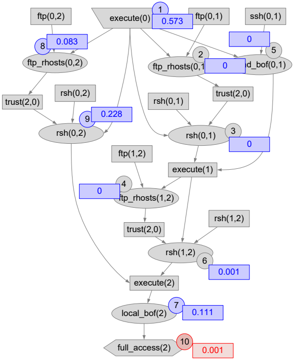
  all amount to humble changes of the Poisson distribution (into something else), i.e., effectively only 1 line of code needs to be changed

More to see, try, discuss at the poster session!

Example 1: Defender can spot-check on $V' = \{1, 2, \ldots, 9\}$ (i.e., everywhere)

- The PBE obtained is in pure strategies, prescribing the defender to periodically patch potential local buffer overflows at machine 2 (optimal pure strategy being `local_bof(2)`), while the attacker is best off by choosing the attack path `execute(0)` $\rightarrow$ `ftp_rhosts(0,2)` $\rightarrow$ `rsh(0,2)` $\rightarrow$ `full_access(2)`.
- This matches the intuition of the best strategy being the defense of the target, by avoiding exploits thereon.
- Since all attack paths intersect at the node `local_bof(2)`, this equilibrium is not surprising.

- The equilibrium utility for the attacker is it to be located at positions $V = \{1, 2, \ldots, 10\}$ with probabilities $U^* \approx (0.573, 0, 0, 0, 0, 0.001, 0.111, 0.083, 0.228, 0.001)$



Example 2: Defender can spot check only on $V' = \{2, 3, 5, 6, 8\}$, i.e., can fix FTP and RSH connections, as well as buffer overflows.

- The equilibrium utility for the attacker is it to be located at positions $V = \{1, 2, \ldots, 10\}$ with probabilities $U^* \approx$ (0.545, 0.017, 0.030, 0.022, 0.012, 0.034, 0.128, 0.021, 0.045, 0.146)