

A simulation-optimization approach for information security risk management

Elmar Kiesling, Andreas Ekelhart, Bernhard Grill,
Christine Strauß, Christian Stummer



International Conference on Operations Research
September 4, 2013; Rotterdam



Agenda

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

Current IT security challenges

- ▶ **Growing complexity** of information systems
 - ▶ **More targeted attacks** by motivated adversaries
 - ▶ **Increasing sophistication** of attacks, exploiting
 - ▶ software vulnerabilities
 - ▶ network vulnerabilities
 - ▶ cognitive biases
 - ▶ insider knowledge and access
 - ▶ etc.
 - ▶ **Heterogeneity of adversaries**
hacktivists, script kiddies, insiders, advanced persistent threats . . .
- **What is the “best” way to mitigate information security risks?**

3 Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

There are no silver bullets

Security is. . .

- ▶ not the result of any particular technical measure
- ▶ not an absolute concept, but involves **tradeoffs**
- ▶ meaningless without specifying a **threat model**
- ▶ a system property that emerges from **interactions**

“Best” solution is highly context-dependent, e.g.,

- ▶ system characteristics
- ▶ threat model
- ▶ available resources
- ▶ decision-makers' risk preferences

4 Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

Problem definition

Objective: choose an “optimal” set of security controls

Approach:

1. Model:
 - ▶ abstract causal dependencies
 - ▶ the information system and its context
 - ▶ adversary behavior
2. Apply control sets and simulate attacks
3. Optimize control sets w.r.t. multiple objectives
4. Support decision-maker in the selection of an efficient control set to implement

Introduction

5 Framework

Knowledge base
Attack patterns
Simulation
Optimization

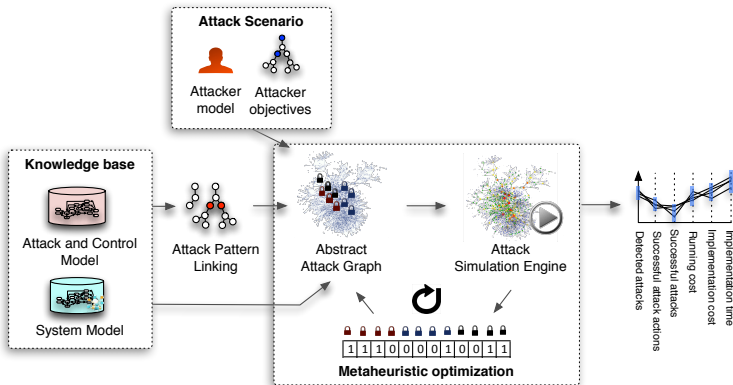
Implementation

Example

Experimental setup
Results

Conclusions

Overview



Introduction

6

Framework

- Knowledge base
- Attack patterns
- Simulation
- Optimization

Implementation

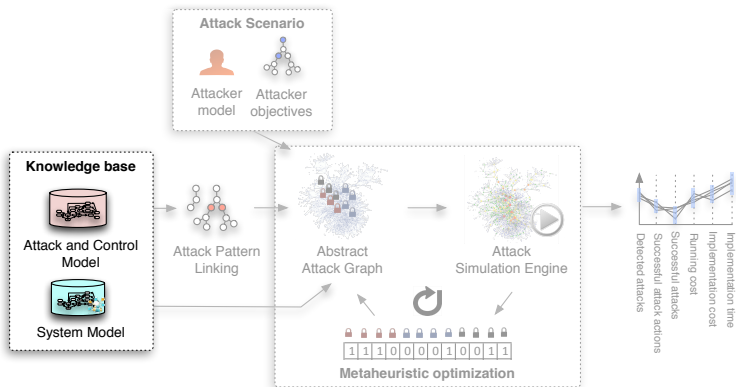
Example

- Experimental setup
- Results

Conclusions

41

Knowledge base



Introduction

Framework

7

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

41

Knowledge base



- ▶ Captures abstract attack knowledge
- ▶ Derived from CAPEC¹

¹<http://capec.mitre.org/>

Introduction

Framework

8 Knowledge base

Attack patterns

Simulation

Optimization

Implementation

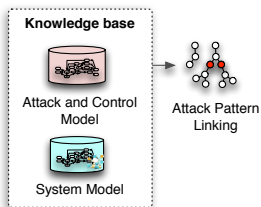
Example

Experimental setup

Results

Conclusions

Attack patterns



Introduction

Framework

Knowledge base

10 **Attack patterns**

Simulation

Optimization

Implementation

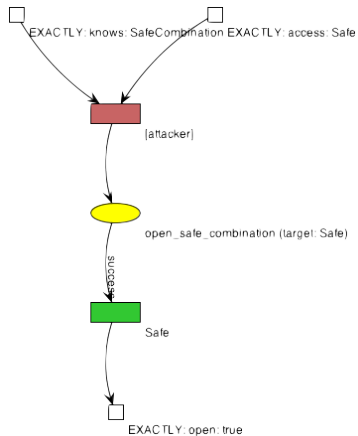
Example

Experimental setup

Results

Conclusions

Attack pattern linking



Introduction

Framework

Knowledge base

11 Attack patterns

Simulation

Optimization

Implementation

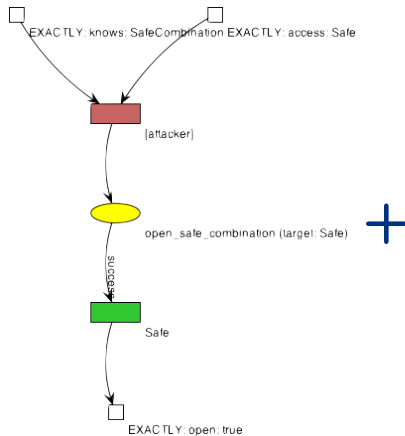
Example

Experimental setup

Results

Conclusions

Attack pattern linking



Introduction

Framework

Knowledge base

11 Attack patterns

Simulation

Optimization

Implementation

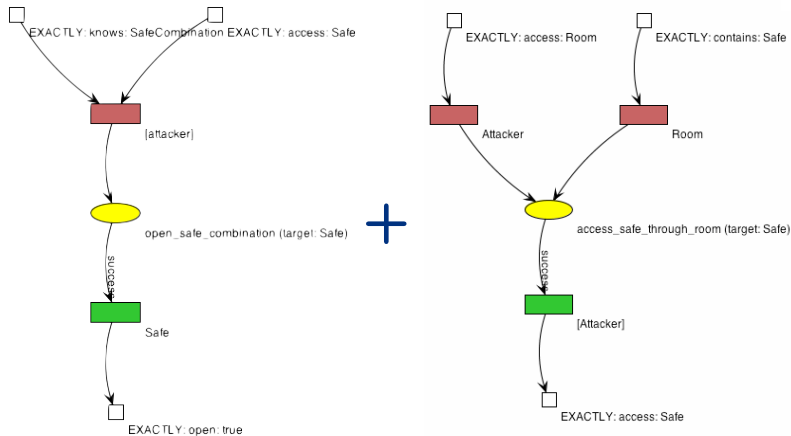
Example

Experimental setup

Results

Conclusions

Attack pattern linking



Introduction

Framework

Knowledge base

11

Attack patterns

Simulation

Optimization

Implementation

Example

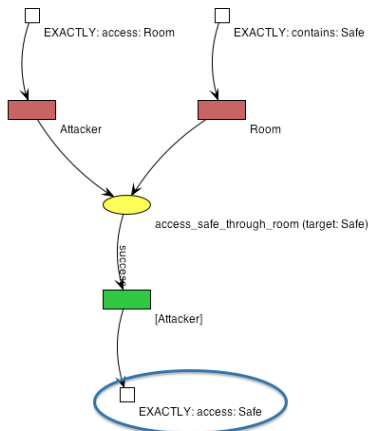
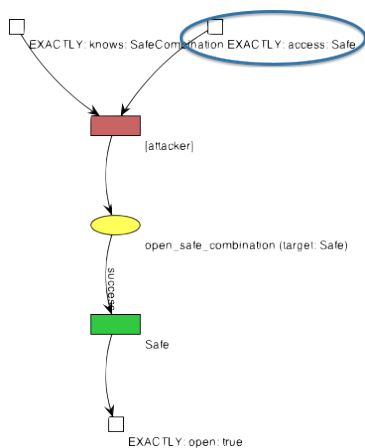
Experimental setup

Results

Conclusions

41

Attack pattern linking



Introduction

Framework

Knowledge base

11

Attack patterns

Simulation

Optimization

Implementation

Example

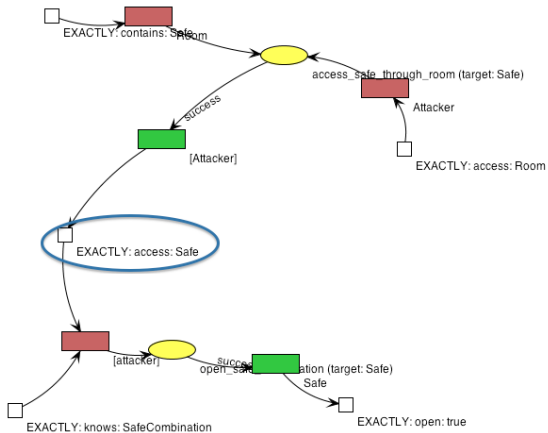
Experimental setup

Results

Conclusions

41

Attack pattern linking



Introduction

Framework

Knowledge base

11 Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

CAPEC [?]

- ▶ Publicly available list of common attack patterns
- ▶ 413 patterns described in varying levels of detail
- ▶ Not fully formalized (textual descriptions)

Transformation:

1. Generic CAPEC pattern → more specific actions
e.g., "134 Email Injection" → `emailKeylogger`, `emailBackdoor`
2. Single CAPEC pattern → sequential atomic actions
e.g., "49 Brute Forcing" → `bruteForce`, `accessHost`, `accessData`
3. Add additional actions
e.g., `accessData`, `accessHost`
4. Formalize
 - ▶ preconditions
 - ▶ postconditions
 - ▶ impact

Introduction

Framework

Knowledge base

12

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

CAPEC example: Brute Force (1)

| Brute Force 1 | | |
|--------------------------------------------------------------------------------|-------------------------------|----------------------|
| Attack Pattern ID: 112 (Standard Attack Pattern Completeness: Complete) | Typical Severity: High | Status: Draft |

Description

Summary

In this attack, some asset (information, functionality, identity, etc.) is protected by a finite secret value. The attacker attempts to gain access to this asset by using trial-and-error to exhaustively explore all the possible secret values in the hope of finding the secret (or a value that is functionally equivalent) that will unlock the asset. Examples of secrets can include, but are not limited to, passwords, encryption keys, database lookup keys, and initial values to one-way functions.

The key factor in this attack is the attacker's ability to explore the possible secret space rapidly. This, in turn, is a function of the size of the secret space and the computational power the attacker is able to bring to bear on the problem. If the attacker has modest resources and the secret space is large, the challenge facing the attacker is intractable. While the defender cannot control the resources available to an attacker, they can control the size of the secret space. Creating a large secret space involves selecting one's secret from as large a field of equally likely alternative secrets as possible and ensuring that an attacker is unable to reduce the size of this field using available clues or cryptanalysis. Doing this is more difficult than it sounds since elimination of patterns (which, in turn, would provide an attacker clues that would help them reduce the space of potential secrets) is difficult to do using deterministic machines, such as computers. Assuming a finite secret space, a brute force attack will eventually succeed. The defender must rely on making sure that the time and resources necessary to do so will exceed the value of the information. For example, a secret space that will likely take hundreds of years to explore is likely safe from raw-brute force attacks.

Introduction

Framework

Knowledge base

13 Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

CAPEC example: Brute Force (2)

Attack Execution Flow

Explore

1. Determine secret testing procedure:

Determine how a potential guess of the secret may be tested. This may be accomplished by comparing some manipulation of the secret to a known value, use of the secret to manipulate some known set of data and determining if the result displays specific characteristics (for example, turning ciphertext into plaintext), or by submitting the secret to some external authority and having the external authority respond as to whether the value was the correct secret. Ideally, the attacker will want to determine the correctness of their guess independently since involvement of an external authority is usually slower and can provide an indication to the defender that a brute-force attack is being attempted.

Attack Step Techniques

| ID | Attack Step Technique Description | Environments |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 1 | Determine if there is a way to parallelize the attack. Most brute force attacks can take advantage of parallel techniques by dividing the search space among available resources, thus dividing the average time to success by the number of resources available. If there is a single choke point, such as a need to check answers with an external authority, the attacker's position is significantly degraded. | env-All |

2. Reduce search space:

Find ways to reduce the secret space. The smaller the attacker can make the space they need to search for the secret value, the greater their chances for success. There are a great many ways in which the search space may be reduced.

Attack Step Techniques

| ID | Attack Step Technique Description | Environments |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| 1 | If possible, determine how the secret was selected. If the secret was determined algorithmically (such as by a random number generator) the algorithm may have patterns or dependencies that reduce the size of the secret space. If the secret was created by a human, behavioral factors may, if not completely reduce the space, make some types of secrets more likely than others. (For example, humans may use the same secrets in multiple places or use secrets that look or sound familiar for ease of recall.) | env-All |
| 2 | If the secret was chosen algorithmically, cryptanalysis can be applied to the algorithm to discover patterns in this algorithm. (This is true even if the secret is not used in cryptography.) Periodicity, the need for seed values, or weaknesses in the generator all can result in a significantly smaller secret space. | env-All |
| 3 | If the secret was chosen by a person, social engineering and simple espionage can indicate patterns in their secret selection. If old secrets can be learned (and a target may feel they have little need to protect a secret that has been replaced) hints as to their selection preferences can be gleaned. These can include character substitutions a target employs, patterns in sources (dates, famous phrases, music lyrics, family members, etc.). Once these patterns have been determined, the initial efforts of a brute-force attack can focus on these areas. | env-All |
| 4 | Some algorithmic techniques for secret selection may leave indicators that can be tested for relatively easily and which could then be used to eliminate large areas of the search space for consideration. For example, it may be possible to determine that a secret does or does not start with a given character after a relatively small number of tests. Alternatively, it might be possible to discover the length of the secret relatively easily. These discoveries would significantly reduce the search space, thus increasing speed with which the attacker discovers the secret. | env-All |

3. Expand victory conditions:

It is sometimes possible to expand victory conditions. For example, the attacker might not need to know the exact secret but simply needs a value that produces the same result using a one-way function. While doing this does not reduce the size of the search space, the presence of multiple victory conditions does reduce the likely amount of time that the attacker will need to explore the space before finding a workable value.

Exploit

1. Gather information so attack can be performed independently.:

If possible, gather the necessary information so a successful search can be determined without consultation of an external authority. This can be accomplished by capturing ciphertext (if the goal is decoding the text) or the encrypted password dictionary (if the goal is learning passwords).

Introduction

Framework

Knowledge base

13 Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

CAPEC example: Brute Force (3)

▼ Attack Prerequisites

The attacker must be able to determine when they have successfully guessed the secret. As such, one-time pads are immune to this type of attack since there is no way to determine when a guess is correct.

▼ Methods of Attack

- Brute Force

▼ Attacker Skills or Knowledge Required

Skill or Knowledge Level: Low

The attack simply requires basic scripting ability to automate the exploration of the search space. More sophisticated attackers may be able to use more advanced methods to reduce the search space and increase the speed with which the secret is located.

▼ Resources Required

Ultimately, the speed with which an attacker discovers a secret is directly proportional to the computational resources the attacker has at their disposal. This attack method is resource expensive: having large amounts of computational power do not guarantee timely success, but having only minimal resources makes the problem intractable against all but the weakest secret selection procedures.

▼ Indicators-Warnings of Attack

Description

Repeated submissions of incorrect secret values may indicate a brute force attack. For example, repeated bad passwords when accessing user accounts or repeated queries to databases using non-existent keys.

Description

Attempts to download files protected by secrets (usually using encryption) may be a precursor to an offline attack to break the file's encryption and read its contents. This is especially significant if the file itself contains other secret values, such as password files.

Description

If the attacker is able to perform the checking offline then there will likely be no indication that an attack is ongoing.

▼ Obfuscation Techniques

Description

The attack is impossible to detect if the attacker can test for successful discovery of the secret value independently, without needing to consult an external authority.

Description

If an external authority must be consulted, the attacker can attempt to space out their guesses to avoid a large number of failed guesses in a short period of time, but doing so slows the attack to the point of making it unworkable against all but the most trivial secret spaces. As such, if an external authority must be consulted the attacked is unlikely to be able to keep the attack secret.

Introduction

Framework

Knowledge base

13 Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

CAPEC example: Brute Force (4)

▼ Solutions and Mitigations

Select a provably large secret space for selection of the secret. Provably large means that the procedure by which the secret is selected does not have artifacts that significantly reduce the size of the total secret space.

Do not provide the means for an attacker to determine success independently. This forces the attacker to check their guesses against an external authority, which can slow the attack and warn the defender. This mitigation may not be possible if testing material must appear externally, such as with a transmitted cryptotext.

▼ Attack Motivation-Consequences

| Scope | Technical Impact | Note |
|-----------------|-----------------------------------|------|
| Confidentiality | Read application data | |
| Confidentiality | Gain privileges / assume identity | |
| Access_Control | | |
| Authorization | | |

▼ Related Weaknesses

| CWE-ID | Weakness Name | Weakness Relationship Type |
|--------|-------------------------------------|----------------------------|
| 330 | Use of Insufficiently Random Values | Secondary |
| 326 | Inadequate Encryption Strength | Secondary |
| 521 | Weak Password Requirements | Secondary |

▼ Related Attack Patterns

| Nature | Type | ID | Name | Description | W |
|-----------|------|-----|------------------------------------------------------------------------|-------------|------|
| ChildOf | Ⓞ | 223 | Probabilistic Techniques | | 1000 |
| HasMember | Ⓞ | 344 | WASC Threat Classification 2.0 - WASC-11 - Brute Force | | 333 |
| ParentOf | Ⓜ | 20 | Encryption Brute Forcing | | 1000 |
| ParentOf | Ⓜ | 49 | Password Brute Forcing | | 1000 |

▼ Relevant Security Requirements

Protect sensitive data, even when the data is encrypted. If an attacker can gain access to encrypted data, they can mount a brute-force attack independently. The defender will not be aware of this attack or be able to do anything about it and at that point it is purely a function of the attacker's available resources as to how long it takes them to learn the secret.

Monitor activity logs for suspicious activity. An attacker that must use an external authority to check their brute-force guesses is easy to detect, but only if that external authority is monitoring activity and detects the abnormally large number of failed guesses.

▼ Related Guidelines

- Do not assume secrets will protect sensitive data in the long-term
- Monitor systems for suspicious activity.

▼ Purposes

- Penetration

Introduction

Framework

Knowledge base

13 Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

Conclusions

Prolog rule formulation: Brute force

Preconditions

```
action_bruteForce(Attacker, TargetHost, TargetGroup):-
    technicalSkillLevel(Attacker, TechnicalSkillLevel),
    TechnicalSkillLevel >= 1,
    owned(Attacker, AttackHost),
    connected(AttackHost, TargetHost, rdpProtocol, rdpPort),
    accessHost(TargetGroup, TargetHost, _),
    not(inGroup(Attacker, TargetGroup)).
```

Postcondition

```
exec_success_action_bruteForce(Attacker, TargetHost, TargetGroup):-
    assert(inGroup(Attacker, TargetGroup)).
```

Impact

```
action_impact(action_bruteForce, confidentiality).
impact_success_bruteForce(Attacker, TargetHost, TargetGroup, SecurityAttribute, Impact):-
    importance(TargetGroup, SecurityAttribute, Impact).
```

Simulation attributes

```
/** cost, time, base probability, maxTries, simultaneous */
action_properties(action_bruteForce, 0, 18000, 0.01, 0, true).
available_action(action_bruteForce).
```

Introduction

Framework

Knowledge base

14

Attack patterns

Simulation

Optimization

Implementation

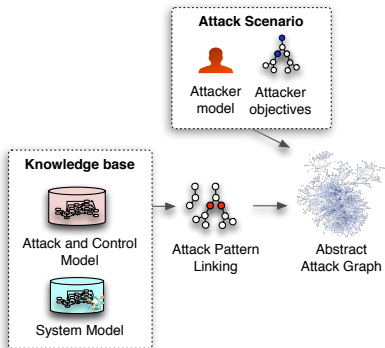
Example

Experimental setup

Results

Conclusions

Simulation



Introduction

Framework

Knowledge base

Attack patterns

15

Simulation

Optimization

Implementation

Example

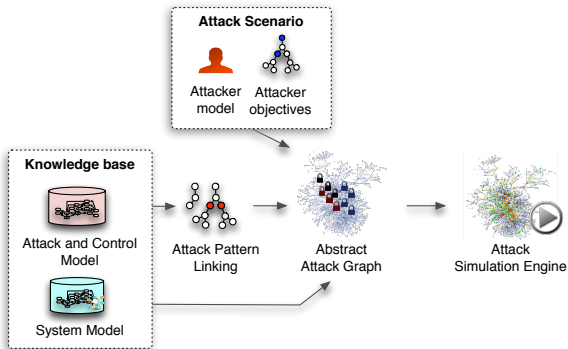
Experimental setup

Results

Conclusions

41

Simulation



Introduction

Framework

- Knowledge base
- Attack patterns

15

Simulation

- Optimization

Implementation

Example

- Experimental setup
- Results

Conclusions

41

Discrete Event Scheduling



Introduction

Framework

Knowledge base

Attack patterns

16 **Simulation**

Optimization

Implementation

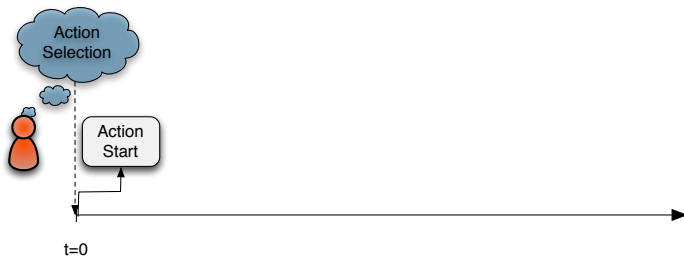
Example

Experimental setup

Results

Conclusions

Discrete Event Scheduling



Introduction

Framework

Knowledge base

Attack patterns

16

Simulation

Optimization

Implementation

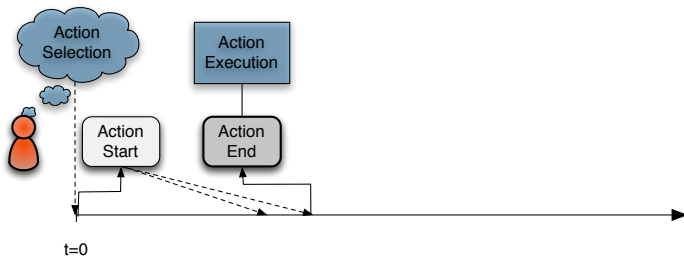
Example

Experimental setup

Results

Conclusions

Discrete Event Scheduling



Introduction

Framework

Knowledge base

Attack patterns

16

Simulation

Optimization

Implementation

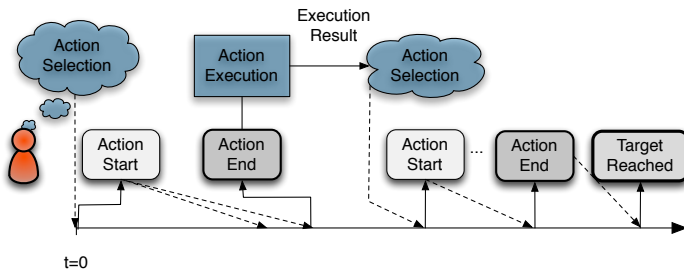
Example

Experimental setup

Results

Conclusions

Discrete Event Scheduling



Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

16

Implementation

Example

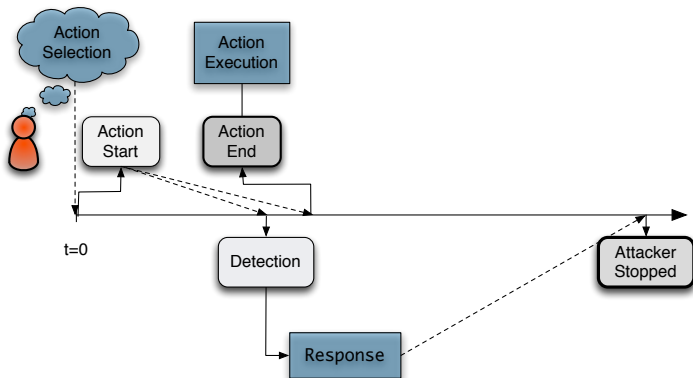
Experimental setup

Results

Conclusions

41

Discrete Event Scheduling



Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

16

Implementation

Example

Experimental setup

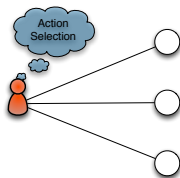
Results

Conclusions

41

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17 Simulation

Optimization

Implementation

Example

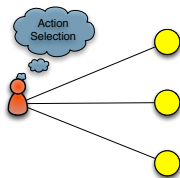
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

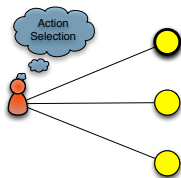
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

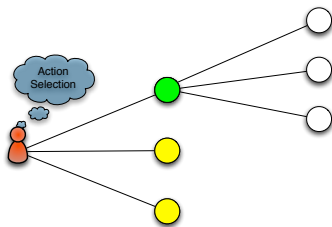
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

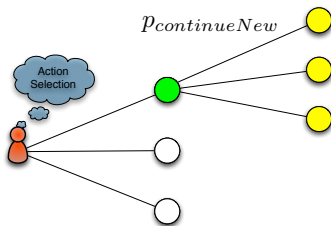
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

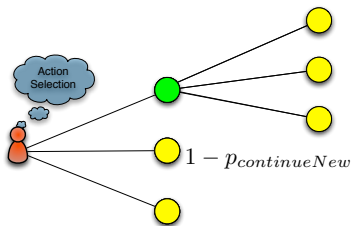
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

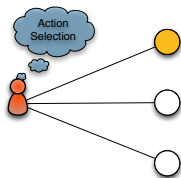
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

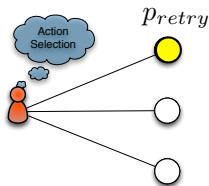
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

Example

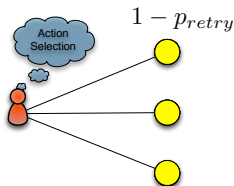
Experimental setup

Results

Conclusions

Behavioral model

Choice set:



Choice function: for all considered actions $a \in A$

1. Calculate distance in abstract graph:

$$d_a^{rel} \leftarrow \frac{d(a,t)}{\max(d(a,t))+1}$$

2. Calculate weight:

$$W_a \leftarrow p_{suc}(a)^{w_{suc}} \left(1 - p_{det}(a)\right)^{w_{det}} \left(1 - d_a^{rel}\right)^{w_{dist}}$$

3. **return** *weightedChoice*(A, W)

Introduction

Framework

Knowledge base

Attack patterns

17

Simulation

Optimization

Implementation

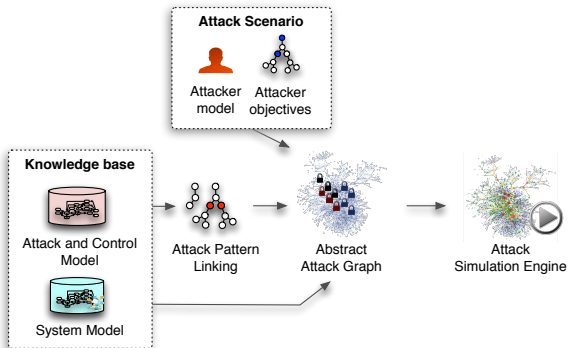
Example

Experimental setup

Results

Conclusions

Optimization



Introduction

Framework

Knowledge base

Attack patterns

Simulation

18

Optimization

Implementation

Example

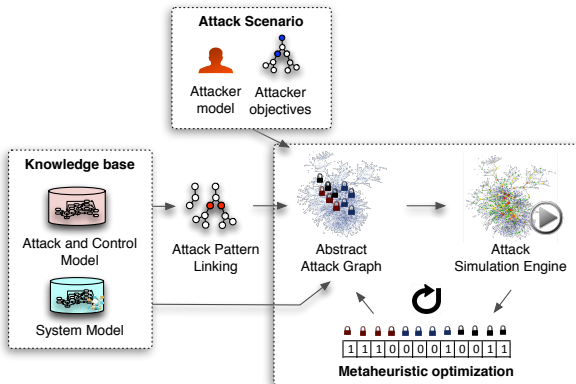
Experimental setup

Results

Conclusions

41

Optimization



Introduction

Framework

Knowledge base

Attack patterns

Simulation

18 Optimization

Implementation

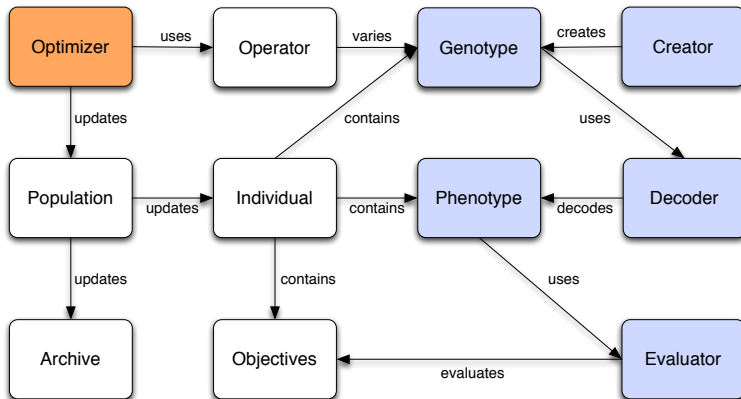
Example

Experimental setup

Results

Conclusions

Implementation: Optimization



Source: adapted from [?]

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

19

Implementation

Example

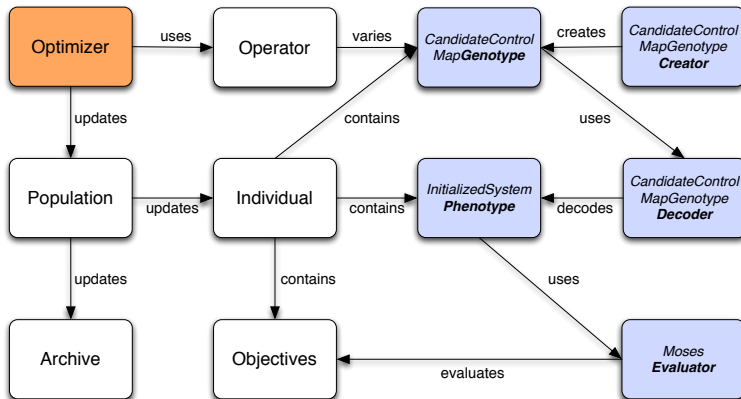
Experimental setup

Results

Conclusions

41

Implementation: Optimization



Source: adapted from [?]

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

19

Implementation

Example

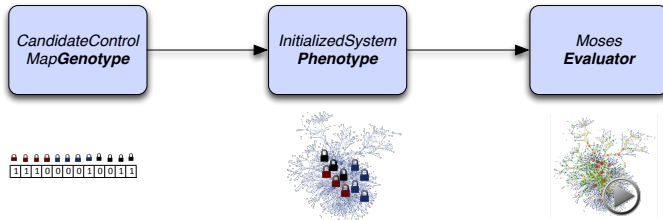
Experimental setup

Results

Conclusions

41

Evaluation of control portfolios



- ▶ Probabilistic → multiple replications per control set
- ▶ Currently reduced to a deterministic problem using expected/median/worst case values etc.

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

20

Implementation

Example

Experimental setup

Results

Conclusions

41

Implementation

Knowledge base

- ▶ Initial experiments with OWL ontologies
- ▶ SWI-Prolog:¹ current rule-based implementation
- ▶ JPL:² Java access

Simulation

- ▶ Java 1.6
- ▶ Mason 14:³ discrete-event core
- ▶ Colt 1.2:⁴ random distributions
- ▶ Jung 2.0.1:⁵ graph structures and visualization
- ▶ Log4j, XStream, JUnit, Commons, ...

Optimization

- ▶ Opt4j 2.7⁶: evolutionary computation framework

¹ <http://www.swi-prolog.org>

² <http://www.swi-prolog.org/packages/jpl>

³ <http://cs.gmu.edu/~eclab/projects/mason/>

⁴ <http://acs.lbl.gov/software/colt/>

⁵ <http://jung.sourceforge.net/>

⁶ <http://opt4j.sourceforge.net/>

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

21 Implementation

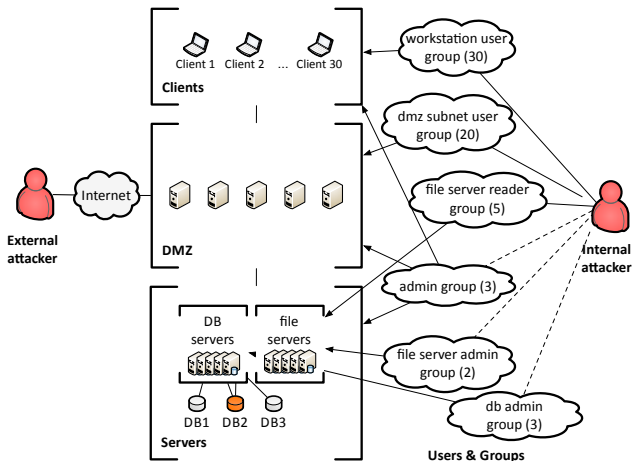
Example

Experimental setup

Results

Conclusions

Scenario domain



Introduction

Framework

- Knowledge base
- Attack patterns
- Simulation
- Optimization

Implementation

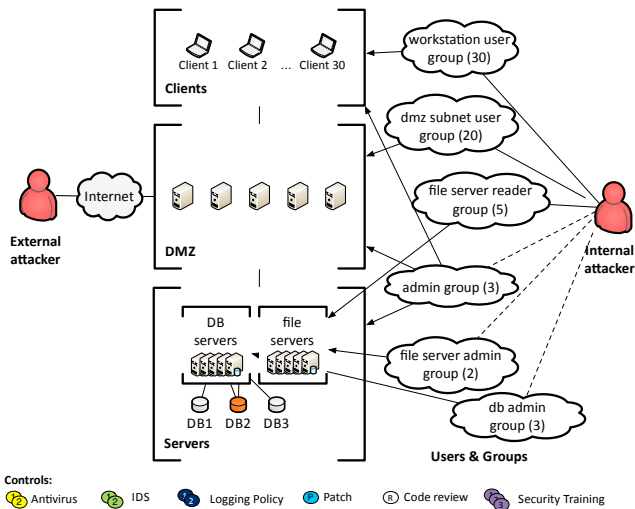
22 Example

- Experimental setup
- Results

Conclusions

41

Scenario domain



Introduction

Framework

Knowledge base
Attack patterns
Simulation
Optimization

Implementation

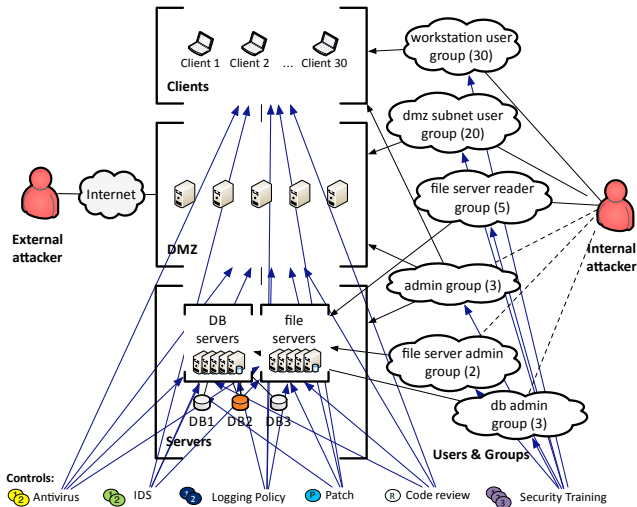
22 Example

Experimental setup
Results

Conclusions

41

Scenario domain



Introduction

Framework

Knowledge base
Attack patterns
Simulation
Optimization

Implementation

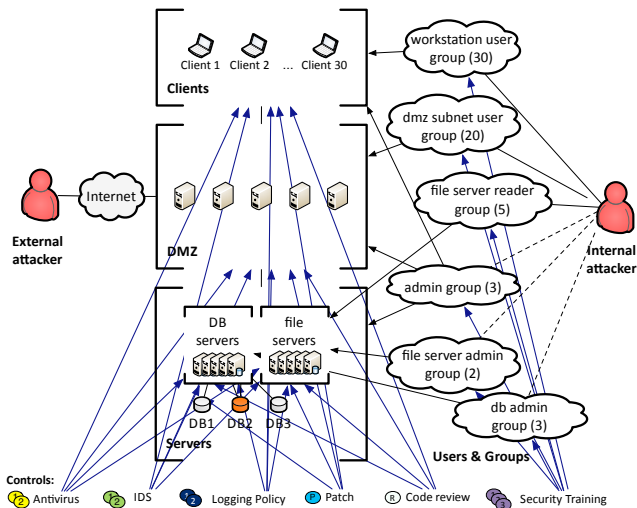
22 Example

Experimental setup
Results

Conclusions

41

Scenario domain



58 binary decision variables (control-asset assignments)

Introduction

Framework

Knowledge base
Attack patterns
Simulation
Optimization

Implementation

22 Example

Experimental setup
Results

Conclusions

41

Adversary types

Characteristics

| | time (mins) | w_{det} | w_{suc} | w_{dist} | access |
|--------------------|-------------|-----------|-----------|------------|--------------|
| Employee | 2500 | 0.45 | 0.25 | 0.30 | workstations |
| Administrator | 5000 | 0.50 | 0.20 | 0.30 | all hosts |
| Skilled External | 3333 | 0.30 | 0.40 | 0.30 | - |
| Unskilled External | 1667 | 0.30 | 0.40 | 0.30 | - |
| APT | ∞ | 0.50 | 0.20 | 0.30 | - |

Available actions (based on skill level, access)

| | |
|---------------------------------------|----------------------|
| Employee (skill: 0) | shoulderSurfing |
| Unskilled external (skill: 1) | spearfish |
| | sqlInjection |
| | socialAttack |
| | bruteForce |
| | emailKeylogger |
| | emailBackdoor |
| Skilled external (skill: 2) | + bufferOverflow |
| | + directoryTraversal |
| Admin (skill: 2) | (all above) |
| Advanced persistent threat (skill: 3) | + zeroDay |

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

23

Experimental setup

Results

Conclusions

41

Optimization objectives

1. Minimize cost of controls
2. Minimize target condition achievement
3. Maximize detection of attacks
4. Minimize confidentiality impact (L/M/H)
5. Minimize integrity impact (L/M/H)
6. Minimize availability impact (L/M/H)

L/M/H: low, medium, high in lexicographic order

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

24

Experimental setup

Results

Conclusions

Parameter settings

Simulation: 50 replications per control set

Optimization: 500 generations

► Population

- $\alpha = 100$ (population size)
- $\mu = 25$ (number of parents per generation)
- $\lambda = 25$ (number of offsprings per generation)
- Initialization: $\vec{1}, \vec{0}$, remaining random
(i.e., each control included with $p = 0.5$)

► **Selection:** NSGA2, 2 tournaments

► **Crossover:** 2-point crossover @ rate 0.95

► **Mutation:** mixed permutation (insert, revert, swap)
rate $1/n$

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

25

Experimental setup

Results

Conclusions

Results

Runtime (3GHz Xeon, currently only single core used)
~ 90 mins (admin) – ~ 50 hrs (APT)

Proposed efficient solutions

- ▶ administrator: 2
- ▶ employee: 58
- ▶ unskilled external: 104
- ▶ skilled external: 306
- ▶ advanced persistent threat: 251

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

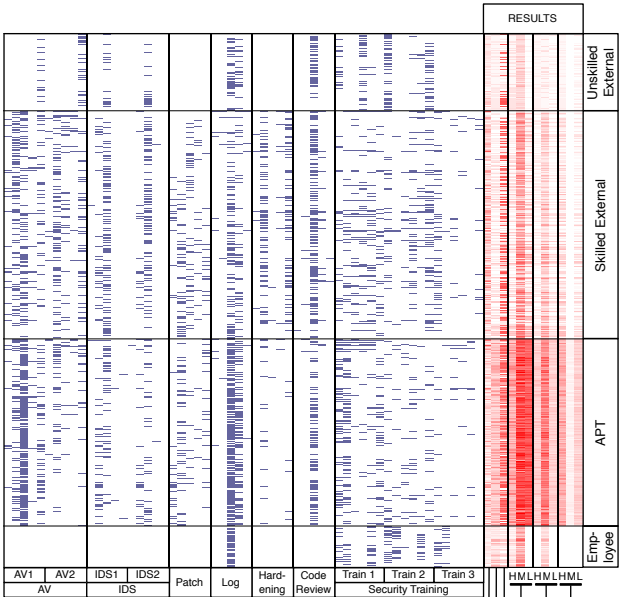
26

Results

Conclusions

41

Results: Overview



Cost
 Target condition reached
 Detected attacks
 Confidentiality impact
 Integrity impact
 Availability impact

Administrator example attack trace



action_accessData

Target:
null

Asset mapping:
TargetHost: dbServerHosts
Data: db2

Results: Administrator

av1 on subnet1Hosts
av1 on empHosts
av1 on dbServerHosts
av1 on fileServerHosts
av1 on workstationHosts
av2 on dbServerHosts
av2 on empHosts
av2 on dbServerHosts
av2 on fileServerHosts
av2 on workstationHosts
ids1 on empHosts
ids1 on dbServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on empHosts
ids2 on dbServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on workstationHosts
patchCVE_2013_04_22 on empHosts
patchCVE_2013_04_22 on dbServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on empHosts
logPolicy1 on dbServerHosts
logPolicy1 on fileServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on empHosts
webServerHardening1 on dbServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on empHosts
codeReview1 on dbServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on dbAdminGroup
securityTraining1 on subnet1UserGroup
securityTraining1 on fileServerUserGroup
securityTraining1 on fileServerUserGroup
securityTraining2 on adminGroup
securityTraining2 on dbAdminGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on fileServerUserGroup
securityTraining2 on workstationUserGroup
securityTraining3 on adminGroup
securityTraining3 on dbAdminGroup
securityTraining3 on subnet1UserGroup
securityTraining3 on fileServerUserGroup
securityTraining3 on workstationUserGroup
securityTraining3 on fileServerUserReaderGroup
securityTraining3 on workstationUserGroup

Cost
Target condition not met
Detected status
Confidentiality high
Confidentiality medium
Confidentiality low
Integrity medium
Integrity high
Availability high
Availability medium
Availability low

Results: Administrator

av1 on subnet1Hosts
av1 on empHosts
av1 on dbServerHosts
av1 on fileServerHosts
av1 on workstationHosts
av2 on empHosts
av2 on dbServerHosts
av2 on fileServerHosts
av2 on workstationHosts
ids1 on empHosts
ids1 on dbServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on empHosts
ids2 on dbServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on workstationHosts
patchCVE_2013_04_22 on empHosts
patchCVE_2013_04_22 on dbServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on empHosts
logPolicy1 on dbServerHosts
logPolicy1 on fileServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on empHosts
webServerHardening1 on dbServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on empHosts
codeReview1 on dbServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on dbAdminGroup
securityTraining1 on subnet1UserGroup
securityTraining1 on fileServerUserGroup
securityTraining1 on fileServerUserReaderGroup
securityTraining2 on adminGroup
securityTraining2 on dbAdminGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on fileServerUserGroup
securityTraining2 on fileServerUserReaderGroup
securityTraining3 on adminGroup
securityTraining3 on dbAdminGroup
securityTraining3 on subnet1UserGroup
securityTraining3 on fileServerUserGroup
securityTraining3 on fileServerUserReaderGroup
securityTraining3 on workstationUserGroup

Target condition reached
Confidentiality high
Confidentiality medium
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low

Target condition always reached

Results: Administrator

Single high confidentiality impact

av1 on subnet1Hosts
av1 on empHosts
av1 on dbServerHosts
av1 on fileServerHosts
av1 on workstationHosts
av2 on empHosts
av2 on dbServerHosts
av2 on fileServerHosts
av2 on workstationHosts
ids1 on empHosts
ids1 on dbServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on empHosts
ids2 on dbServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on empHosts
patchCVE_2013_04_22 on dbServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on empHosts
logPolicy1 on dbServerHosts
logPolicy1 on fileServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on empHosts
webServerHardening1 on dbServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on empHosts
codeReview1 on dbServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on dbAdminGroup
securityTraining1 on subnet1UserGroup
securityTraining1 on fileServerUserGroup
securityTraining1 on fileServerUserReaderGroup
securityTraining2 on adminGroup
securityTraining2 on dbAdminGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on fileServerUserGroup
securityTraining2 on fileServerUserReaderGroup
securityTraining3 on adminGroup
securityTraining3 on dbAdminGroup
securityTraining3 on subnet1UserGroup
securityTraining3 on fileServerUserGroup
securityTraining3 on fileServerUserReaderGroup
securityTraining3 on workstationUserGroup
Cost

Target condition reached

Confidentiality high

Integrity medium

Availability low

Availability high

Availability medium

Availability low

Results: Administrator

av1 on subnet1Hosts
av1 on empHosts
av1 on dbServerHosts
av1 on fileServerHosts
av2 on workstationHosts
av2 on empHosts
av2 on dbServerHosts
av2 on fileServerHosts
ids1 on workstationHosts
ids1 on empHosts
ids1 on dbServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on workstationHosts
ids2 on dbServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on workstationHosts
patchCVE_2013_04_22 on empHosts
patchCVE_2013_04_22 on dbServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on subnet1Hosts
logPolicy1 on dbServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on empHosts
webServerHardening1 on dbServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on empHosts
codeReview1 on dbServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on dbAdminGroup
securityTraining1 on subnet1UserGroup
securityTraining1 on fileServerUserGroup
securityTraining1 on fileServerUserGroup
securityTraining2 on adminGroup
securityTraining2 on dbAdminGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on fileServerUserGroup
securityTraining2 on workstationUserGroup
securityTraining3 on adminGroup
securityTraining3 on dbAdminGroup
securityTraining3 on subnet1UserGroup
securityTraining3 on fileServerUserGroup
securityTraining3 on workstationUserGroup
securityTrainings on workstationUserGroup
Cost

Detected attacks
Confidentiality medium
Integrity medium
Availability low

Log policy improves detection

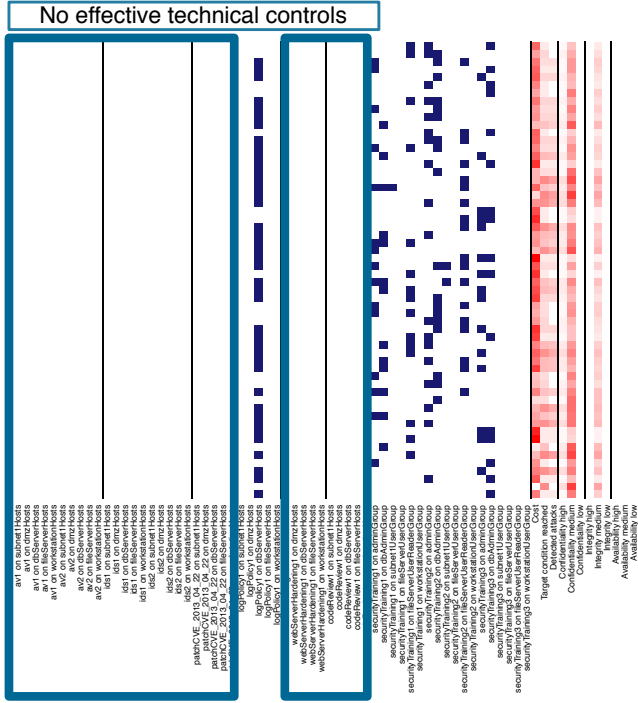
Employee example attack trace



Results: Employee

| | |
|--------------------------------------------|--|
| av1 on subnet11Hosts | |
| av1 on dsServerHosts | |
| av1 on dsServerHosts | |
| av1 on fileServerHosts | |
| av1 on workstationHosts | |
| av2 on subnet11Hosts | |
| av2 on dsServerHosts | |
| av2 on dsServerHosts | |
| av2 on fileServerHosts | |
| av2 on workstationHosts | |
| ids1 on subnet11Hosts | |
| ids1 on dsServerHosts | |
| ids1 on dsServerHosts | |
| ids1 on fileServerHosts | |
| ids1 on workstationHosts | |
| ids2 on subnet11Hosts | |
| ids2 on dsServerHosts | |
| ids2 on dsServerHosts | |
| ids2 on fileServerHosts | |
| ids2 on workstationHosts | |
| patchCVE_2013_04_22 on subnet11Hosts | |
| patchCVE_2013_04_22 on dsServerHosts | |
| patchCVE_2013_04_22 on dsServerHosts | |
| patchCVE_2013_04_22 on fileServerHosts | |
| patchCVE_2013_04_22 on workstationHosts | |
| logPolicy1 on dirzHosts | |
| logPolicy1 on dsServerHosts | |
| logPolicy1 on dsServerHosts | |
| logPolicy1 on fileServerHosts | |
| logPolicy1 on workstationHosts | |
| webServerHardening1 on dsServerHosts | |
| webServerHardening1 on dsServerHosts | |
| webServerHardening1 on dsServerHosts | |
| webServerHardening1 on fileServerHosts | |
| webServerHardening1 on workstationHosts | |
| codeReview1 on dirzHosts | |
| codeReview1 on dsServerHosts | |
| codeReview1 on dsServerHosts | |
| codeReview1 on fileServerHosts | |
| codeReview1 on workstationHosts | |
| securityTraining1 on abAdminGroup | |
| securityTraining1 on subnet1UserGroup | |
| securityTraining1 on fileServerUserGroup | |
| securityTraining1 on fileServerReaderGroup | |
| securityTraining2 on adminGroup | |
| securityTraining2 on abAdminGroup | |
| securityTraining2 on subnet1UserGroup | |
| securityTraining2 on subnet1UserGroup | |
| securityTraining2 on fileServerUserGroup | |
| securityTraining2 on fileServerReaderGroup | |
| securityTraining3 on adminGroup | |
| securityTraining3 on abAdminGroup | |
| securityTraining3 on subnet1UserGroup | |
| securityTraining3 on fileServerUserGroup | |
| securityTraining3 on fileServerReaderGroup | |
| securityTraining3 on workstationUserGroup | |
| Target condition was | |
| Detected attacks | |
| Confidentiality high | |
| Confidentiality medium | |
| Confidentiality low | |
| Integrity high | |
| Integrity medium | |
| Integrity low | |
| Availability high | |
| Availability medium | |
| Availability low | |

Results: Employee



Results: Employee

Security trainings are effective

| | |
|------------------------------------------|--|
| av1 on subnet11Hosts | |
| av1 on dsServerHosts | |
| av1 on fileServerHosts | |
| av1 on workstationHosts | |
| av2 on subnet11Hosts | |
| av2 on dsServerHosts | |
| av2 on fileServerHosts | |
| av2 on workstationHosts | |
| ids1 on subnet11Hosts | |
| ids1 on dsServerHosts | |
| ids1 on fileServerHosts | |
| ids1 on workstationHosts | |
| ids2 on subnet11Hosts | |
| ids2 on dsServerHosts | |
| ids2 on fileServerHosts | |
| ids2 on workstationHosts | |
| patchCVE_2013_04_22 on subnet11Hosts | |
| patchCVE_2013_04_22 on dsServerHosts | |
| patchCVE_2013_04_22 on fileServerHosts | |
| patchCVE_2013_04_22 on workstationHosts | |
| logPolicy1 on dirzHosts | |
| logPolicy1 on dsServerHosts | |
| logPolicy1 on fileServerHosts | |
| logPolicy1 on workstationHosts | |
| webServerHardening1 on dirzHosts | |
| webServerHardening1 on dsServerHosts | |
| webServerHardening1 on fileServerHosts | |
| webServerHardening1 on workstationHosts | |
| codeReview1 on dirzHosts | |
| codeReview1 on dsServerHosts | |
| codeReview1 on fileServerHosts | |
| codeReview1 on workstationHosts | |
| securityTraining1 on adminGroup | |
| securityTraining1 on subnet11UserGroup | |
| securityTraining1 on fileServerUserGroup | |
| securityTraining1 on dsServerUserGroup | |
| securityTraining2 on adminGroup | |
| securityTraining2 on adminGroup | |
| securityTraining2 on subnet11UserGroup | |
| securityTraining2 on fileServerUserGroup | |
| securityTraining2 on dsServerUserGroup | |
| securityTraining3 on adminGroup | |
| securityTraining3 on adminGroup | |
| securityTraining3 on subnet11UserGroup | |
| securityTraining3 on fileServerUserGroup | |
| securityTraining3 on dsServerUserGroup | |
| Cost | |
| Target condition was | |
| Detected attacks | |
| Confidentiality high | |
| Confidentiality medium | |
| Confidentiality low | |
| Integrity high | |
| Integrity medium | |
| Integrity low | |
| Availability high | |
| Availability medium | |
| Availability low | |

Results: Employee

Success rate can be reduced from 46% to 6%

av1 on subnet11Hosts
av1 on dsServerHosts
av1 on dsServerHosts
av1 on fileServerHosts
av1 on workstationHosts
av2 on subnet11Hosts
av2 on dsServerHosts
av2 on dsServerHosts
av2 on fileServerHosts
av2 on workstationHosts
ids1 on subnet11Hosts
ids1 on dsServerHosts
ids1 on dsServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on subnet11Hosts
ids2 on dsServerHosts
ids2 on dsServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on subnet11Hosts
patchCVE_2013_04_22 on dsServerHosts
patchCVE_2013_04_22 on dsServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on dsServerHosts
logPolicy1 on dsServerHosts
logPolicy1 on dsServerHosts
logPolicy1 on fileServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on dsServerHosts
webServerHardening1 on dsServerHosts
webServerHardening1 on dsServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on dsServerHosts
codeReview1 on dsServerHosts
codeReview1 on dsServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on adminGroup
securityTraining1 on adminGroup
securityTraining1 on subnet1UserGroup
securityTraining1 on fileServerReaderGroup
securityTraining1 on fileServerReaderGroup
securityTraining2 on adminGroup
securityTraining2 on adminGroup
securityTraining2 on adminGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on subnet1UserGroup
securityTraining2 on fileServerReaderGroup
securityTraining2 on workstationUserGroup
securityTraining3 on adminGroup
securityTraining3 on adminGroup
securityTraining3 on fileServerReaderGroup
securityTraining3 on fileServerReaderGroup
securityTraining3 on workstationUserGroup
securityTraining3 on workstationUserGroup

Target condition reached

Confidentiality high
Confidentiality medium
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low

Results: Employee

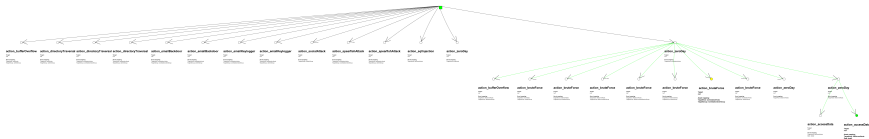
Log policy can increase detection rate to ~ 1/3

```
av1 on subnet11hosts
av1 on subnet11hosts
av1 on dBServerHosts
av1 on fileServerHosts
av1 on workstationHosts
av2 on subnet11hosts
av2 on subnet11hosts
av2 on dBServerHosts
av2 on fileServerHosts
av2 on workstationHosts
ids1 on subnet11hosts
ids1 on subnet11hosts
ids1 on dBServerHosts
ids1 on fileServerHosts
ids1 on workstationHosts
ids2 on subnet11hosts
ids2 on subnet11hosts
ids2 on dBServerHosts
ids2 on fileServerHosts
ids2 on workstationHosts
patchCVE_2013_04_22 on subnet11hosts
patchCVE_2013_04_22 on dBServerHosts
patchCVE_2013_04_22 on fileServerHosts
patchCVE_2013_04_22 on workstationHosts
logPolicy1 on subnet11hosts
logPolicy1 on dBServerHosts
logPolicy1 on fileServerHosts
logPolicy1 on workstationHosts
webServerHardening1 on dirzHosts
webServerHardening1 on dBServerHosts
webServerHardening1 on fileServerHosts
webServerHardening1 on workstationHosts
codeReview1 on dirzHosts
codeReview1 on dBServerHosts
codeReview1 on fileServerHosts
codeReview1 on workstationHosts
securityTraining1 on adminGroup
securityTraining1 on subset1UserGroup
securityTraining1 on fileServerUserGroup
securityTraining1 on fileServerReaderGroup
securityTraining2 on adminGroup
securityTraining2 on subset1UserGroup
securityTraining2 on subset2UserGroup
securityTraining2 on fileServerUserGroup
securityTraining2 on fileServerReaderGroup
securityTraining3 on adminGroup
securityTraining3 on subset1UserGroup
securityTraining3 on subset2UserGroup
securityTraining3 on fileServerUserGroup
securityTraining3 on fileServerReaderGroup
Cost
```

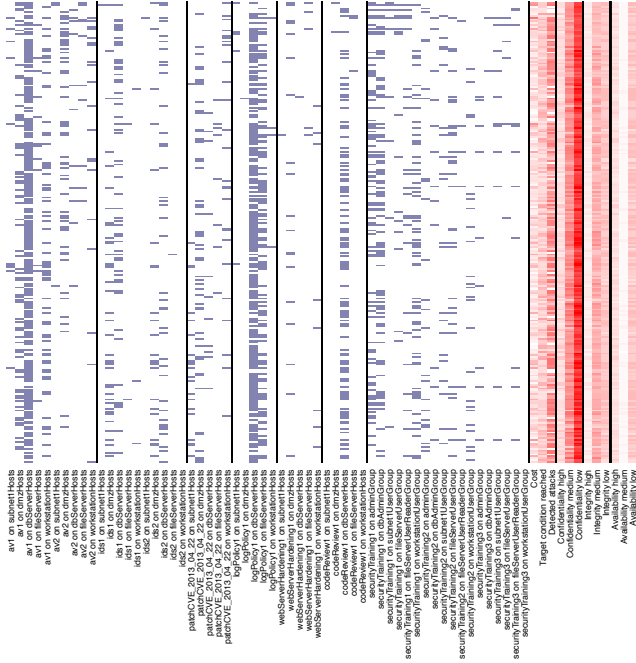
Detected attacks

| | |
|-----------------|--------|
| Confidentiality | medium |
| Confidentiality | low |
| Integrity | medium |
| Integrity | low |
| Availability | high |
| Availability | medium |
| Availability | low |

APT example attack trace



Results: Advanced persistent threat



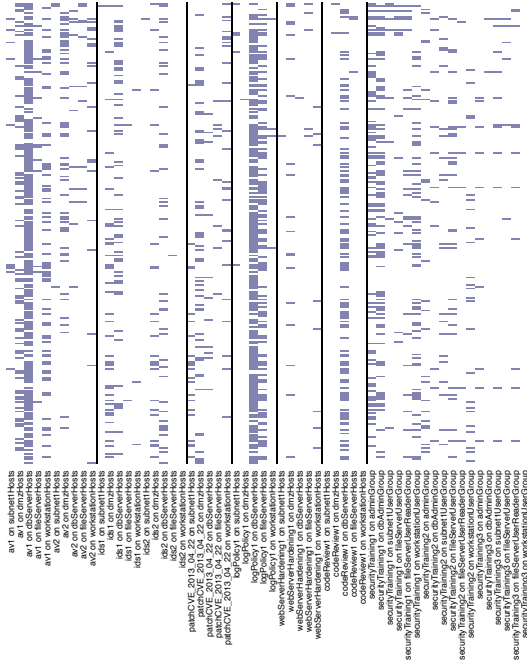
Results: Advanced persistent threat

Wide range of effective controls

av1 on subnet1-hosts
av1 on dmz-hosts
av1 on db-Serverhosts
av1 on webserver-hosts
av1 on workstation-hosts
av2 on subnet1-hosts
av2 on dmz-hosts
av2 on db-Serverhosts
av2 on webserver-hosts
av2 on workstation-hosts
ids1 on subnet1-hosts
ids1 on dmz-hosts
ids1 on db-Serverhosts
ids1 on webserver-hosts
ids1 on workstation-hosts
ids2 on subnet1-hosts
ids2 on dmz-hosts
ids2 on db-Serverhosts
ids2 on webserver-hosts
ids2 on workstation-hosts
patchCVE_2013_04_22 on subnet1-hosts
patchCVE_2013_04_22 on dmz-hosts
patchCVE_2013_04_22 on db-Serverhosts
patchCVE_2013_04_22 on webserver-hosts
patchCVE_2013_04_22 on workstation-hosts
logPolicy1 on subnet1-hosts
logPolicy1 on dmz-hosts
logPolicy1 on db-Serverhosts
logPolicy1 on webserver-hosts
logPolicy1 on workstation-hosts
webServerHardening1 on subnet1-hosts
webServerHardening1 on dmz-hosts
webServerHardening1 on db-Serverhosts
webServerHardening1 on webserver-hosts
webServerHardening1 on workstation-hosts
codeReview1 on subnet1-hosts
codeReview1 on dmz-hosts
codeReview1 on db-Serverhosts
codeReview1 on webserver-hosts
codeReview1 on workstation-hosts
securityTraining1 on admin-Group
securityTraining1 on subnet1-User-Group
securityTraining1 on db-Admin-Group
securityTraining1 on db-Server-Reader-Group
securityTraining1 on workstation-User-Group
securityTraining2 on admin-Group
securityTraining2 on subnet1-User-Group
securityTraining2 on db-Admin-Group
securityTraining2 on db-Server-Reader-Group
securityTraining2 on workstation-User-Group
securityTraining3 on admin-Group
securityTraining3 on subnet1-User-Group
securityTraining3 on db-Admin-Group
securityTraining3 on db-Server-Reader-Group

Target condition reached
Derecord attach
Confidentiality high
Confidentiality medium
Confidentiality low
Integrity low
Integrity medium
Integrity high
Availability high
Availability medium
Availability low

Results: Advanced persistent threat



Target condition reached

Confidentiality high
Confidentiality medium
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low

Results: Advanced persistent threat

IDS and log policies can raise detection rates to ~ 2/3

```
av1 on subnet1-ho5ts
av1 on dmz-ho5ts
av1 on ds-Serverho5ts
av1 on ds-Serverho5ts
av1 on workstation-ho5ts
av2 on subnet1-ho5ts
av2 on dmz-ho5ts
av2 on ds-Serverho5ts
av2 on ds-Serverho5ts
av2 on workstation-ho5ts
ias1 on dmz-ho5ts
ias1 on ds-Serverho5ts
ias1 on ds-Serverho5ts
ias1 on workstation-ho5ts
ias2 on dmz-ho5ts
ias2 on ds-Serverho5ts
ias2 on ds-Serverho5ts
ias2 on workstation-ho5ts

patchCVE_2013_04_22 on dmz-ho5ts
patchCVE_2013_04_22 on ds-Serverho5ts
patchCVE_2013_04_22 on ds-Serverho5ts
patchCVE_2013_04_22 on workstation-ho5ts

logpolicy1 on subnet1-ho5ts
logpolicy1 on dmz-ho5ts
logpolicy1 on ds-Serverho5ts
logpolicy1 on ds-Serverho5ts
logpolicy1 on workstation-ho5ts

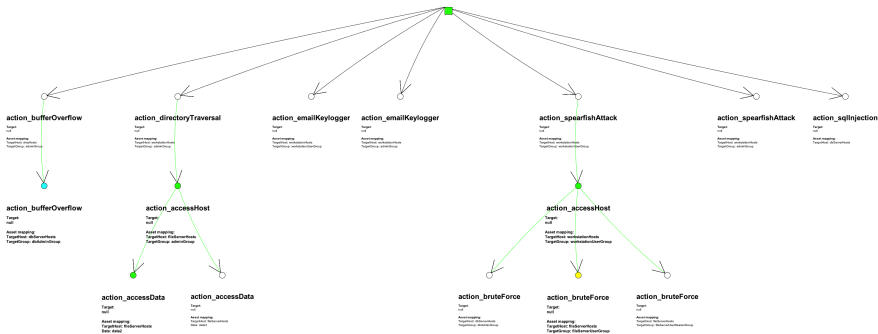
webServerHardening1 on dmz-ho5ts
webServerHardening1 on ds-Serverho5ts
webServerHardening1 on ds-Serverho5ts
webServerHardening1 on workstation-ho5ts

codeReview1 on subnet1-ho5ts
codeReview1 on dmz-ho5ts
codeReview1 on ds-Serverho5ts
codeReview1 on ds-Serverho5ts
codeReview1 on workstation-ho5ts

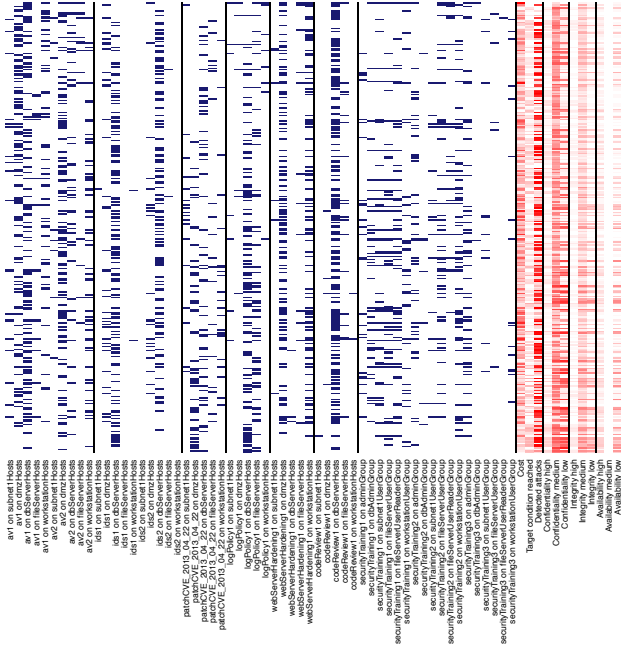
securityTraining1 on admin-Group
securityTraining1 on subnetUser-Group
securityTraining1 on subnetUser-Group
securityTraining1 on ds-ServerUser-Reader-Group
securityTraining1 on ds-ServerUser-Reader-Group
securityTraining1 on workstationUser-Group
securityTraining1 on workstationUser-Group
securityTraining2 on admin-Group
securityTraining2 on admin-Group
securityTraining2 on ds-ServerUser-Reader-Group
securityTraining2 on ds-ServerUser-Reader-Group
securityTraining2 on workstationUser-Group
securityTraining2 on workstationUser-Group
securityTraining3 on subnetUser-Group
securityTraining3 on subnetUser-Group
securityTraining3 on ds-ServerUser-Reader-Group
securityTraining3 on ds-ServerUser-Reader-Group
securityTraining3 on workstationUser-Group
securityTraining3 on workstationUser-Group
```

Target condition reached
Confidentiality high
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low

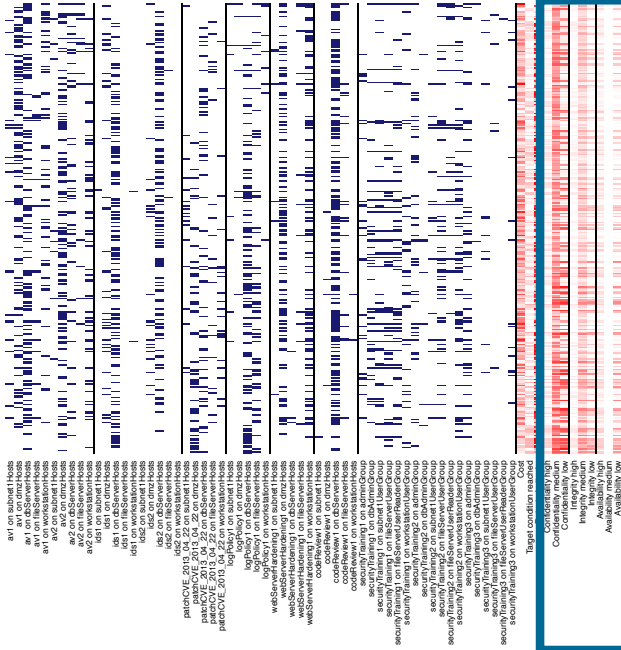
Skilled external example attack trace



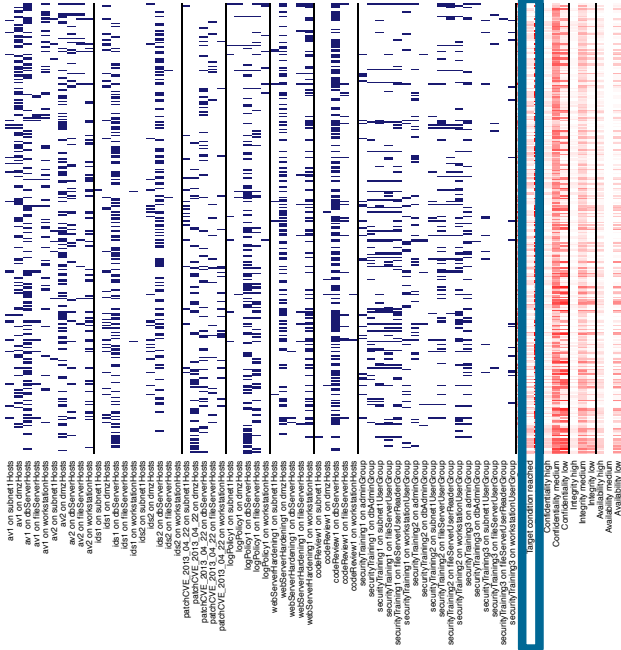
Results: Skilled external



Results: Skilled external



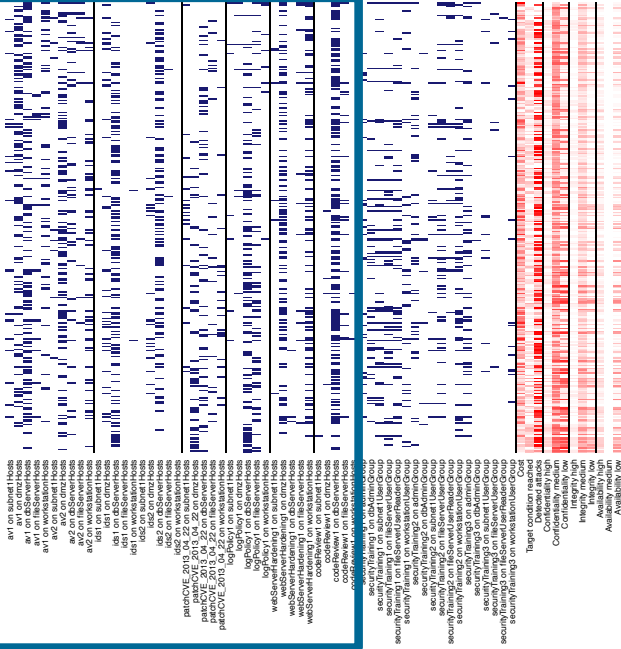
Results: Skilled external



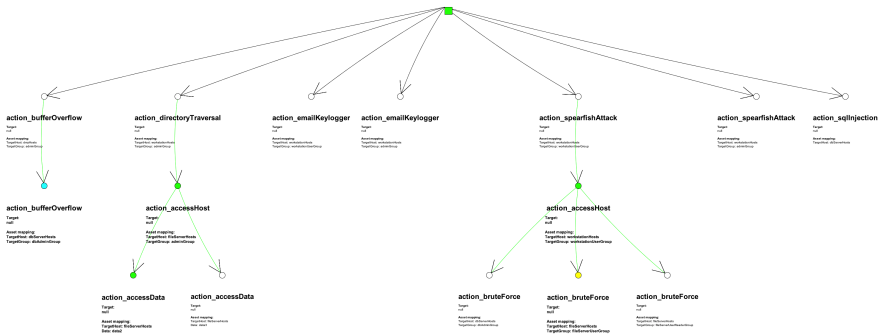
Lower success probability

Results: Skilled external

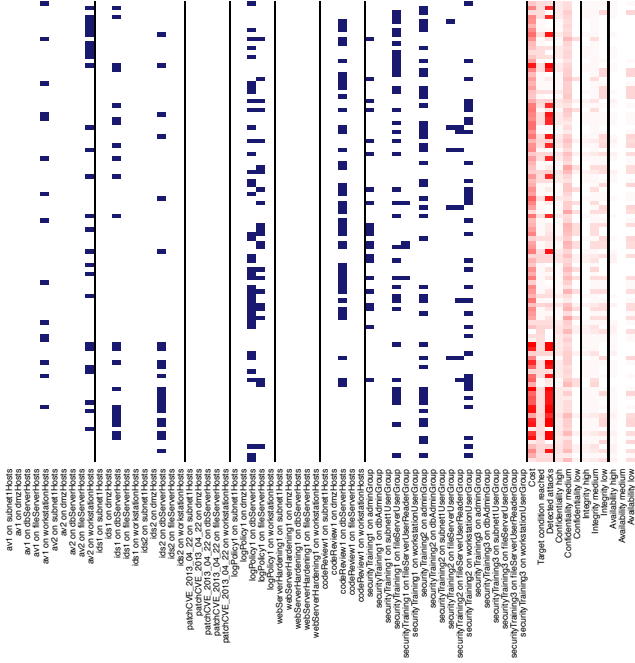
More effective technical controls



Unskilled external example attack trace



Results: Unskilled external

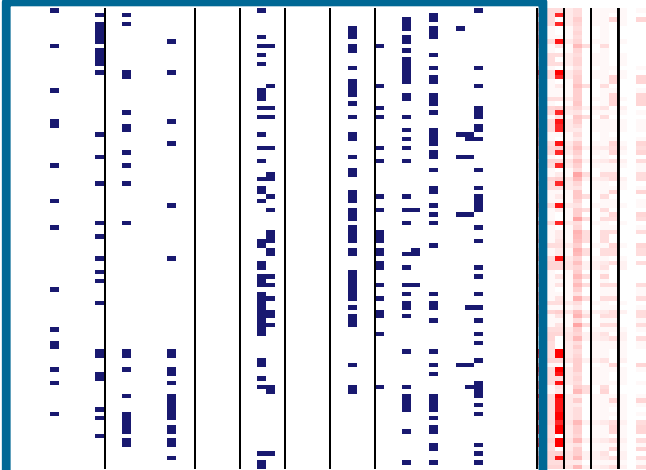


Results: Unskilled external

Fewer technical controls

av1 on subnet1/Hosts
av1 on crmz/Hosts
av1 on dbServer/Hosts
av1 on erp/Hosts
av1 on workstation/Hosts
av2 on subnet1/Hosts
av2 on crmz/Hosts
av2 on dbServer/Hosts
av2 on erp/Hosts
av2 on workstation/Hosts
ids1 on subnet1/Hosts
ids1 on crmz/Hosts
ids1 on dbServer/Hosts
ids1 on erp/Hosts
ids1 on workstation/Hosts
ids2 on subnet1/Hosts
ids2 on crmz/Hosts
ids2 on dbServer/Hosts
ids2 on erp/Hosts
ids2 on workstation/Hosts
logPolicy1 on crmz/Hosts
logPolicy1 on dbServer/Hosts
logPolicy1 on erp/Hosts
logPolicy1 on workstation/Hosts
webServerHardening1 on subnet1/Hosts
webServerHardening1 on dbServer/Hosts
webServerHardening1 on erp/Hosts
webServerHardening1 on workstation/Hosts
codeReview1 on subnet1/Hosts
codeReview1 on crmz/Hosts
codeReview1 on dbServer/Hosts
codeReview1 on erp/Hosts
codeReview1 on workstation/Hosts
securityTraining1 on admin/Group
securityTraining1 on subnet1/UserGroup
securityTraining1 on workstation/UserGroup
securityTraining1 on fileServer/UserGroup
securityTraining1 on fileServer/ReaderGroup
securityTraining2 on admin/Group
securityTraining2 on crmz/Group
securityTraining2 on dbServer/Group
securityTraining2 on fileServer/UserGroup
securityTraining2 on fileServer/ReaderGroup
securityTraining3 on admin/Group
securityTraining3 on crmz/Group
securityTraining3 on dbServer/UserGroup
securityTraining3 on fileServer/ReaderGroup
securityTraining3 on workstation/UserGroup

Target condition reached
Detected attacks
Confidentiality high
Confidentiality medium
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low



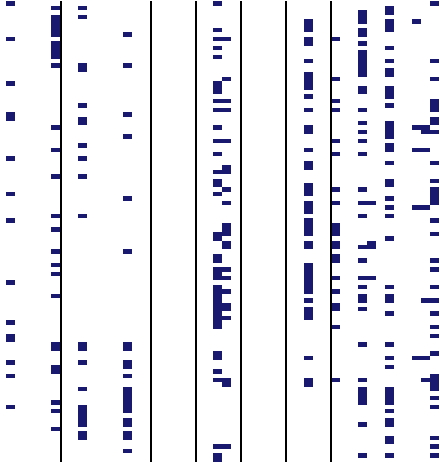
Results: Unskilled external

av1 on subnet1/Hosts
av1 on ctmz/Hosts
av1 on dbServer/Hosts
av1 on workstation/Hosts
av1 on workstation/Hosts
av2 on subnet1/Hosts
av2 on ctmz/Hosts
av2 on dbServer/Hosts
av2 on workstation/Hosts
av2 on workstation/Hosts
ids1 on subnet1/Hosts
ids1 on ctmz/Hosts
ids1 on dbServer/Hosts
ids1 on workstation/Hosts
ids1 on workstation/Hosts
ids2 on subnet1/Hosts
ids2 on ctmz/Hosts
ids2 on dbServer/Hosts
ids2 on workstation/Hosts
ids2 on workstation/Hosts
patchCVE_2013_04_22 on subnet1/Hosts
patchCVE_2013_04_22 on ctmz/Hosts
patchCVE_2013_04_22 on dbServer/Hosts
patchCVE_2013_04_22 on workstation/Hosts
patchCVE_2013_04_22 on workstation/Hosts
logPolicy1 on ctmz/Hosts
logPolicy1 on dbServer/Hosts
logPolicy1 on workstation/Hosts
logPolicy1 on workstation/Hosts
logPolicy1 on workstation/Hosts
webServerHardening1 on subnet1/Hosts
webServerHardening1 on ctmz/Hosts
webServerHardening1 on dbServer/Hosts
webServerHardening1 on workstation/Hosts
webServerHardening1 on workstation/Hosts
codeReview1 on subnet1/Hosts
codeReview1 on ctmz/Hosts
codeReview1 on dbServer/Hosts
codeReview1 on workstation/Hosts
codeReview1 on workstation/Hosts
securityTraining1 on admin/Group
securityTraining1 on subnet1/UserGroup
securityTraining1 on ctmz/UserGroup
securityTraining1 on fileServer/UserGroup
securityTraining1 on workstation/UserGroup
securityTraining1 on workstation/UserGroup
securityTraining2 on admin/Group
securityTraining2 on ctmz/UserGroup
securityTraining2 on fileServer/UserGroup
securityTraining2 on workstation/UserGroup
securityTraining2 on workstation/UserGroup
securityTraining3 on admin/Group
securityTraining3 on ctmz/UserGroup
securityTraining3 on fileServer/UserGroup
securityTraining3 on workstation/UserGroup
securityTraining3 on workstation/UserGroup

Target condition reached

Confidentiality high
Confidentiality medium
Confidentiality low
Integrity high
Integrity medium
Integrity low
Availability high
Availability medium
Availability low

Success probability can be lowered to ~ 3%



Conclusions

Summary

- ▶ Simulation-Optimization framework for IT security
- ▶ Attacker-centric approach

Current research challenges

- ▶ Knowledge base: attack pattern formalization
- ▶ Simulation: cognitive and behavioral model
- ▶ Optimization:
 - ▶ cost of portfolio evaluations
 - ▶ cost of permutations

Future work

- ▶ Control selection → system design
(very large design space + constraints)
- ▶ Problem-specific genotype structure
- ▶ Interactive analysis and decision-support

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

38 Conclusions

41

Q & A

Contact:
ekiesling@sba-research.org

Introduction

Framework

Knowledge base

Attack patterns

Simulation

Optimization

Implementation

Example

Experimental setup

Results

39 Conclusions

Part I
Appendix

References I