An Intelligent Board of Security Countermeasure Cases in Prolog.

Frank Appiah

An Intelligent Board of Storytelling of Castle of Fortress in Prolog.

Prof Frank Appiah AKA FAEng PhD (KCL)
Email:appiahnsiahfrank@gmail.com
10.2020.

Abstract. This shows how to use logical program made from declarative semantics of interpretable language of prolog is used to make an intelligent board director able to come to a decision on security prevention in Vunerabilities at Office.

Keywords. AI, intelligent, storyboard, logic, program, symbolic, telling.

1 Introduction.

The international dimensions of counter -feiting by duplication, thefting by deletion replacement or insertion replacement, cyberattacking, hacking / cracking on internet / decentralized network is strong gating with incorrect measure, ungaurding on access breach, unlawful entry, uncontrolling access system codes, momenting by passby fights, intern replacement unverifiables and unvalidating information are exploitations that needs to be addressed.

Vulnerability is a weakness in the security system. A threat is blocked by control of vulnerability.

A control is a protective measure used as an action, procedure or technique. Simply, this security measure research report is addressing the following:

- Creation of security controls in unceiled secret information.
- Laying out risk of unceiled secret information and ways of dealing with it.
- Certain on ways of document process - sing with digitized image water- marking.
- Middle aging of counterfeiting by duplication with deletion replacement and insertion replacement.

- A decentralized network with marginal error on control printing with water marking process.
- A counterattack measure in validating and verification of authentic document.

If it is possible or necessary watermarking secret Ceil should be used to prevent ruining access control. Then it should be used.

If counterfeiting by duplication creates a methods of recovery in risky information. Then, it should be recover after incident.

If a security officer or engineer can address duplication copy in cases an attacker deletes and insert a counterfeit copy to be used by the document marker thereby making information lose confidentiality or integrity. Then, it should be engineered for counter measuring. If in a decentralized scenario, the document marker will be able to authenticate as usual to able to have access to the digital document to make a copy for further processing. Then, it should create authenticated access. If the technique of authentication and authorization can be by password or biometry (fingers, iris, height etc). Then it should create technology for culturing and socializing the security process of characterization. If file transfer protocol gives the decentralized manner of network access with secure means. Then it should create confidentiality and availability in the security process. If Unceil secret paper creates vulnerabilities and embarrassment in ruining the authenticity of document. Then it should leave the security room of vulnerabilities. If security agents unchase theft document in a vulnerable situation. Then it should be way to dismissal from the work place.

If it is hard and difficult to physically timestamp all documents at a security site. Then watermarking by stamping should be the way to countermeasure. If vulnerabilities prevention is a means to countermeasure a counterfeit information.

Then finally a Ceil by watermarking should be used.

2 Director Assessment.

A security director or officer or engineer addressing duplication copy in cases an

attacker deletes and insert a counterfeit copy to be used by the document marker thereby making information lose confidentiality or integrity.

In the middle ages of counterfeiting by duplication, a copy of existing image is kept with the security officer or engineer on deletion replacement or insertion replacement. In the castle of counterfeiting by duplication, a different but approved image is quickly inserted into the document processing of the watermarking paper. Then it is casted into decentralized networks with a marginal error on the previous information dissemination from the control printer software. The fortress of counterfeiting by duplication a security officer will counterattack with an invalid document fight in the sense of seizing and requesting a reprint of information to process new.

Director will initiate the main rule of the xProlog application which is coded as below:

```
main :-
 1.  nl,
 2.  write('Security Director Program.'), nl,
 3.  write('>   Enter a selection followed by a period.'), nl,
 4.  write('>   1. Yes, countermeasure'), nl,
 5.  write('>   2. Exit'), nl, nl,
 6.  read(Choice),
 7.  assess_opt(Choice),
 8.  main.

initialization('main').
```

It will first move the cursor to a newline, coded as blue. It will display a text "Security Director Program" on the monitor, coded on line 2 and a newline is called. A selection is read to be assessed with the assess_opt rule of the program on line 7. The assessment is a forward chain of cdd abbreviation rules, count of 12. After each rule has successfully run then the next cdd rule and final cdd then runs the main rule again. This is how the security director works. It identifies each Vulnerability and ask you to counter messure until all assessments are done at the office.
A successful execution run will show as follows:

```
Security Director Program.
>    Enter a selection followed by a period.
>    1. Yes, countermeasure
>    2. Exit
```

This prolog application is a mobile application that a security officer or engineer can use to run security routine with a team in countermeasure strategies. The logic program is programmed in XProlog Android on Honor model from Huawei corporation. The benchmark set from AI Expert on this device is shown in Appendix 3.

Further Reading

[1] Frank Appiah (2020). Security Controls or Countermeasures: Vunerabilities Prevention, Easychair Preprint 4410.
[2] XProlog (2020). XProlog Android Package, Playstore. Programming IDE. Online Accessed.
[3] Android Mobile (2020). Huawei Device Honor Model. Programming with Footprint Minicomputers. Huawei Corporation.

Appendix 1.

Prolog Code.

```
assess_opt(1) :- write('Yes, there is a countermeasure to the exploitation'), nl,
    write('>   Select a direction followed by a period.'), nl,
    write('>   1. Counterfeit by Duplication(CDD)'), nl,
    write('>   2. Countermeasure on Duplication(CDD)'), nl,
    write('>   3. Exit'), nl, nl,
     read(Choice),
     cddtheft(Choice).

cddtheft('1') :-
    write('Identify thefting by  deletion replacement.'), nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on thefting by deletion'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cddrep(Choice),
    main(Choice), halt.

cddrep(1) :-
    write('Identify insertion replacement at office.'),nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on insertion replacement'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd3(Choice),
    main(Choice).

cdd3(1) :-
    write('Identify cyberattack on network at office or home.'),nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on cyberattacking.'), nl,
```

```prolog
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd4(Choice),
    main(Choice).

cdd4(1) :- write('Identify hacking /cracking on internet/decentralized network.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, strong countermeasure on hacking/cracking.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd5(Choice),
    main(Choice).

cdd5(1) :- write('Identify if it is strong  gating with incorrect measure.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on strong gating.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd6(Choice),
    main(Choice).

cdd6(1) :- write('Identify if ungaurding on access breach.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure by gaurding on access breach.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd6(Choice),
    main(Choice).

cdd7(1) :- write('Identify if there is an unlawful entry.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on strong gating.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd8(Choice),
    main(Choice).

cdd8(1) :- write('Identify if it is an uncontrolling access system codes.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on controlling access system codes.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd9(Choice),
    main(Choice).

cdd9(1) :- write('Momenting by passby fights is recalled.'),nl,
write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure on passby riot is checked.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    cdd_10(Choice),
    main(Choice).

cdd_10(1) :-
    write('Identify if it caused by intern replacement unverifiables and unvalidatiables.'),nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure by checking on intern replacement.'), nl,
    write('>   2. Exit'), nl, nl,
```

```prolog
    read(Choice),
    cdd_11(Choice),
    main(Choice).

cdd_11(1) :-
    write('Identify if it is unvalidating information'),nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure check on invalid information.'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    main(Choice).

cdd(_) :-
    write('Unknown operation').


main(3) :- main.

main(2) :- main.

main :-
    nl,
    write('Security Director Program.'), nl,
    write('>   Enter a selection followed by a period.'), nl,
    write('>   1. Yes, countermeasure'), nl,
    write('>   2. Exit'), nl, nl,
    read(Choice),
    assess_opt(Choice), main.

initialization('main').
```

Appendix 2: Execution Runs.

```
message(warning.                    ,       [style_warning(multifile,
(initialization)/1), file_name('/storage/emulated/0/Download/main.pl')]).
message(warning, [style_warning(discontiguous, (initialization)/1),
file_name('/storage/emulated/0/Download/main.pl')]).
message(warning, [style_warning(multifile, main/0),
file_name('/storage/emulated/0/Download/main.pl')]).
message(warning, [style_warning(discontiguous, main/0),
file_name('/storage/emulated/0/Download/main.pl')]).


Security Director Program.
>    Enter a selection followed by a period.
>    1. Yes, countermeasure
>    2. Exit

1.
Yes, there is a countermeasure to the exploitation
>    Select a direction followed by a period.
>    1. Counterfeit by Duplication(CDD)
>    2. Countermeasure on Duplication(CDD)
>    3. Exit
```

```
1.
Identify thefting by  deletion replacement.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on thefting by deletion
>   2. Exit


1.
Identify insertion replacement at office.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on insertion replacement
>   2. Exit


1.
Identify cyberattack on network at office or home.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on cyberattacking.
>   2. Exit


1.
Identify hacking /cracking on internet/decentralized network.
>   Enter a selection followed by a period.
>   1. Yes, strong countermeasure on hacking/cracking.
>   2. Exit


1.
Identify if it is strong  gating with incorrect measure.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on strong gating.
>   2. Exit


1.
Identify if ungaurding on access breach.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure by gaurding on access breach.
>   2. Exit


1.
Identify if there is an unlawful entry.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure checing on lawful entrance.
>   2. Exit


1.
Identify if it is an uncontrolling access system codes.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on controlling access system codes.
>   2. Exit


1.
Momenting by passby fights is recalled.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure on passby riot is checked.
```

```
>   2. Exit

1.
Identify if it caused by intern replacement unverifiables and
unvalidatiables.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure by checking on intern replacement.
>   2. Exit

1.
Identify if it is unvalidating information
>   Enter a selection followed by a period.
>   1. Yes, countermeasure check on invalid information.
>   2. Restart

2.

Security Director Program.
>   Enter a selection followed by a period.
>   1. Yes, countermeasure
>   2. Exit
```

Appendix 3. Benchmark Set from AI Expert magazine.

Benchmark	Iterations	Average
tail_call_atom_atom	50,000	6.20
binary_call_atom_atom	50,000	9.60
cons_list	50,000	6.80
walk_list	50,000	5.20
walk_list_rec	50,000	6.60
args(1)	50,000	6.20
args(2)	50,000	8.80
args(4)	50,000	14.80
args(8)	50,000	25.20
args(16)	50,000	46.40
cons_term	50,000	8.40
walk_term	50,000	7.40
walk_term_rec	50,000	7.00
shallow_backtracking	50,000	5.20
deep_backtracking	50,000	17.80
trail_variables	50,000	15.40
medium_unify	50,000	0.80
deep_unify	10,000	1.00
integer_add	10,000	9.00
floating_add	10,000	10.00
arg(1)	50,000	22.80

arg(2)	50,000	23.20
arg(4)	50,000	20.60
arg(8)	50,000	18.40
arg(16)	50,000	18.60
index	20,000	8.50
assert_unit	10,000	98.00
access_unit	10,000	92.00
slow_access_unit	10,000	94.00
setof	10,000	43.00
pair_setof	10,000	67.00
double_setof	10,000	676.00
bagof	10,000	27.00

28,110 msec