

توضیحات پروژه EvaluatingCSA

ابتدا ساخت مدل را توضیح می دهیم سپس کد ها را روی مدل ها اعمال می کنیم.

مدل

مدل از سه بخش BPMN، Network و Attack تشکیل شده است.

BPMN

در این بخش باید فرآیند ها در یک کسب و کار مشخص شود همچنین ارتباط بین آن ها و مأموریت های مربوطه به همراه میزان اهمیت هر کدام مشخص شود.

بخش BPMN به 4 بخش ResourcePools، Processes، WorkFlows و Missions تقسیم می شود.

ResourcePools

در این بخش مشخص می شود که به چند Resource Pool نیاز داریم اسم هر کدام از آن ها چیست و هر کدام از آن ها چند Resource دارند و آیا بین Resource Pool ها وابستگی وجود دارد یا خیر و همینطور هر Resource Pool به کدام Subnet در بخش Network متصل می شود.

در شکل زیر مثال این بخش را می بینید.

```
YML model1.yml x
1 BPMN:
2 ResourcePools:
3   ResourcePoolNumbers: 4
4   ResourcePool1:
5     Name: "DB"
6     ResourceNumbers: 1
7     Resource1:
8       Name: "DB"
9     Dependencies: []
10    SubnetNumber: 4
11   ResourcePool2:
12     Name: "Servers"
13     ResourceNumbers: 2
14     Resource1:
15       Name: "Server1"
16     Resource2:
17       Name: "Server2"
18     Dependencies: [ DB ]
19     SubnetNumber: 2
20   ResourcePool3:
21     Name: "NginxServer"
22     ResourceNumbers: 1
23     Resource1:
24       Name: "NGINX"
25     Dependencies: []
26     SubnetNumber: 1
27   ResourcePool4:
28     Name: "SMTPServer"
29     ResourceNumbers: 1
30     Resource1:
31       Name: "SMTP"
32     Dependencies: [ Servers ]
33     SubnetNumber: 3
34 Processes: <6 keys>
51 WorkFlows: <2 keys>
72 Missions: <4 keys>
84
```

Processes

در این بخش تمام فرآیندها معرفی می شوند و این که هر کدام از آن ها مربوط به کدام Resource Pool هستند نیز معلوم می شود.

WorkFlows

در این بخش ارتباط فرآیندها با یکدیگر مشخص می شود. می توان چندین Path داشته باشیم. در هر Path می توان لیستی از Processes ها و همچنین GateWays ها داشته باشیم. باید ترتیب اجرا را نیز با کلید OrderedKeys داشته باشیم. در هر Path باید مشخص باشد که Priority که میزان اهمیت آن Path را نشان می دهد وجود دارد یا خیر. در GateWays ها نیز می توان چندین Path داشت.

Priority باعث می شود که میزان اهمیت آن Path برای صاحب کسب و کار مشخص شود.

در شکل زیر مثالی از بخش های Processes و WorkFlows قرار داده شده است.

```

1  model1.yml
2  ResourcePools: <5 keys>
34 Processes:
35   ProcessNumbers: 5
36   Process1:
37     Name: "ReceiveRequest"
38     ResourcePool: NginxServer
39   Process2:
40     Name: "Authenticate"
41     ResourcePool: Servers
42   Process3:
43     Name: "EmailUser"
44     ResourcePool: SMTPServer
45   Process4:
46     Name: "ProcessData"
47     ResourcePool: Servers
48   Process5:
49     Name: "SendResponse"
50     ResourcePool: NginxServer
51 WorkFlows:
52   PathNumbers: 1
53   Path1:
54     HasPriority: False
55     OrderedKeys: [Processes1, GateWays, Processes2]
56     Processes1: [ "Start", ReceiveRequest, Authenticate ]
57     GateWays:
58       Type: "Parallel" #Exclusive and Inclusive
59       Condition: None
60       PathNumbers: 2
61       Path1:
62         HasPriority: True
63         OrderedKeys: [Processes1]
64         Processes1: [ ProcessData ]
65         Priority: 5
66       Path2:
67         HasPriority: True
68         OrderedKeys: [Processes1]
69         Processes1: [ EmailUser ]
70         Priority: 2
71     Processes2: [ SendResponse, "End" ]

```

Missions

در این بخش به هر ماموریت نامی را می دهیم و هر ماموریت از تعدادی فرآیند تشکیل شده است. هر ماموریت نیز Priority دارد.

مجموع تمام Priority این بخش به همراه Priority های بخش WorkFlows عدد Business Importance را تشکیل می دهند.

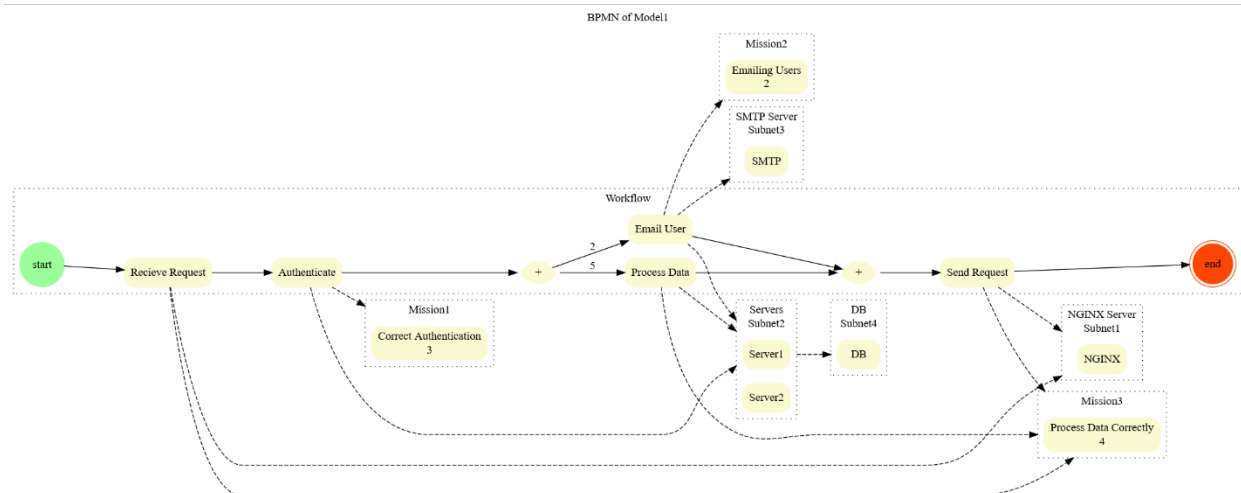
می توان اهمیت هر فرآیند را نیز حساب کرد بدین صورت که اهمیت هر ماموریت بین فرآیند های مربوط به آن تقسیم می شود. این عمل برای Path های درون WorkFlows نیز تکرار می شود یعنی اهمیت هر Path بین فرآیند های مربوطه تقسیم می شود. در نتیجه ما Process Importance را به ازای هر فرآیند داریم. * (در بخش اصلاحات توضیح بیشتر آورده شده است) دو بخش Business Importance و Process Importance درون کد ها محاسبه می شوند. در شکل زیر مثالی از بخش Missions را مشاهده می کنید.

```

1  BPMN:
2  ResourcePools: <5 keys>
34 Processes: <6 keys>
51 WorkFlows: <2 keys>
72 Missions:
73   MissionNumbers: 3
74   Mission1:
75     Name: "Correct Authentication"
76     Processes: [Authenticate]
77     Priority: 3
78   Mission2:
79     Name: "Emailing Users"
80     Processes: [ EmailUser ]
81     Priority: 2
82   Mission3:
83     Name: "Process data correctly"
84     Processes: [ ReceiveRequest, ProcessData, SendResponse]
85     Priority: 4
86

```

در شکل زیر بخش BPMN را به صورت تبدیل شده به گراف مشاهده می کنید. (این بخش در پروژه لازم نیست و فقط برای مشاهده بهتر ایجاد شده است)



Network

در این بخش تعداد Subnet ها و اینکه هر Subnet چند Host دارد و اینکه بین کدام Subnet ها ارتباط وجود دارد (ارتباط از اینترنت به کدام Subnet وجود دارد) بیان می شود.

اینکه هر کدام از Host ها چه آدرسی دارند (عدد اول شماره Subnet و عدد دوم شماره Host)، چه سیستم عاملی در آن وجود دارد و چه سرویس ها و Process هایی در آن اجرا می شوند نیز آورده شده است.

این تنظیمات با توجه به ارتباطاتی که در بخش BPMN داریم باید نوشته شود.

به ازای هر Host عددی را به عنوان SecurityFactor بین صفر تا یک قرار می دهیم که نشاندهنده میزان امن بودن سیستم در برابر تهدید است. عدد نزدیک به یک به معنای مقاوم تر بودن است.

می توان فرمولی به عنوان نسبت تعداد آسیب پذیری هایی که Host می داند و می تواند جلوی خرابی ناشی از آن را بگیرد تقسیم بر تمام آسیب پذیری هایی که وجود دارد را برای این عدد پیشنهاد دهیم اما با این کار باعث می شویم که عواملی که در مدل به آن پرداخته نشده مانند تاثیر آنتی ویروس ها، IDS ها و ... را در نظر نگیریم.

به ازای هر Host با توجه به اهمیت هر فرآیند و ارتباط آن فرآیند ابتدا با ResourcePool مربوطه و سپس با Subnet مربوطه، می توان اهمیت هر کدام را به دست آورد. این کار در کد انجام می شود. * (در بخش اصلاحات توضیح بیشتر آورده می شود)

برای هر Host چهار متغیر دودویی در نظر گرفته ایم.

1- IsCompromised که در ابتدا False است.

2- IsCompromisedCompletely که در ابتدا False است. اگر حمله کننده بتواند به

دسترسی root یا administration دسترسی پیدا کند. این متغیر True می شود.

3- `isDataLeaked` که در ابتدا `False` است. اگر اطلاعاتی از `Host` برداشته شود `True` می شود. می تواند در دو حالتی که `IsCompromised`، `True` باشد و زمانی که `IsCompromisedCompletely` هم `True` باشد در نظر گرفته شود.

4- `IsTerminated` اگر حمله کننده تصمیم بگیرد که کلا کارایی یک `Host` را نابود کند آنگاه این متغیر `True` می شود. البته ابتدا `IsCompromisedCompletely` باید `True` باشد.

به ازای 5 حالتی که برای هر `Host` وجود دارد ضرابی را داریم که از اهمیت آن `Host` برای ارزیابی کم می کند.

1- در حالتی که `Host`، `Terminate` شده باشد، تمام ارزش آن `Host` از بین می رود پس ضریب 1 را در نظر می گیریم.

2- زمانی که `Host` کاملاً `Compromised` شده باشد و دیتایی نیز استخراج شده باشد ضریب 0.8 است. یعنی بعد از آن حمله ارزش آن `Host`، 0.2 حالت اصلی می شود.

3- زمانی که `Host`، `Compromised` شده باشد و دیتایی نیز استخراج شده باشد ضریب 0.6 است.

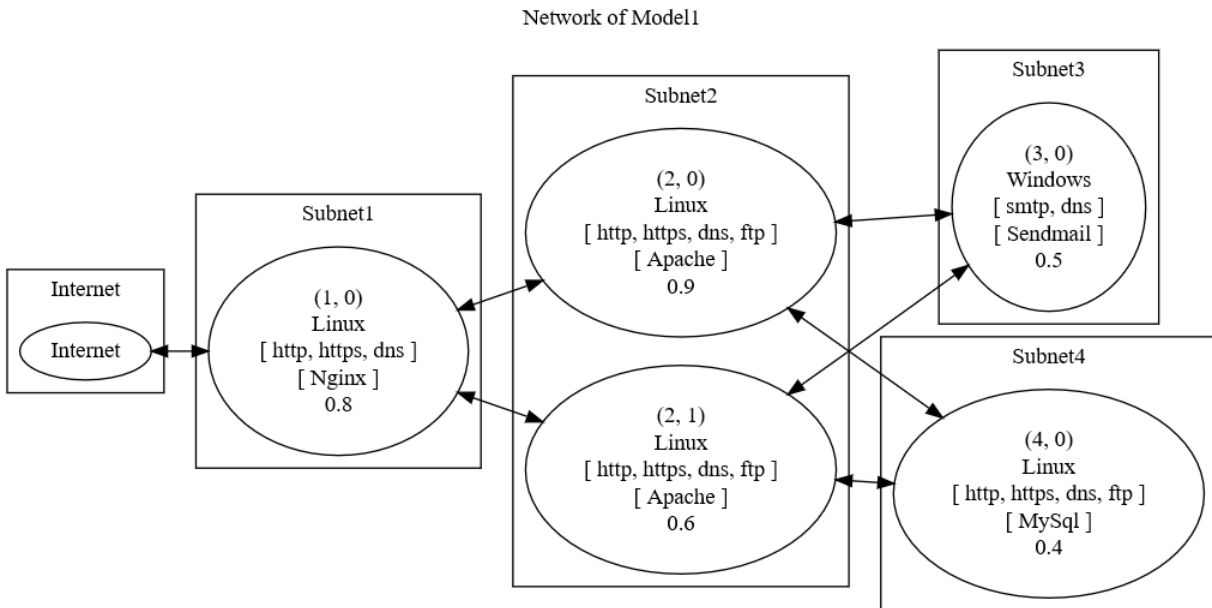
4- زمانی که `Host` کاملاً `Compromised` شده باشد ضریب 0.5 است.

5- زمانی که `Host`، `Compromised` شده باشد ضریب 0.3 است.

در شکل زیر مثالی از بخش `Network` را مشاهده می کنید.


```
YML model1.yml x
1  BPMN: <4 keys>
86
87  Network:
88    SubnetsNumbers: 4
89    Subnets: [ 1, 2, 1, 1 ]
90    Topology:
91      Internet: [Subnet1]
92      Subnet1: [Internet, Subnet2]
93      Subnet2: [Subnet1, Subnet3, Subnet4]
94      Subnet3: [Subnet2]
95      Subnet4: [Subnet2]
96    HostConfiguration:
97      (1,0):
98        Os: linux
99        Services: [ http, https, dns ]
100        Processes: [ Nginx ]
101        SecurityFactor: 0.8
102      (2,0):
103        Os: linux
104        Services: [ http, https, dns, ftp ]
105        Processes: [ Apache ]
106        SecurityFactor: 0.9
107      (2,1):
108        Os: linux
109        Services: [ http, https, dns, ftp ]
110        Processes: [ Apache ]
111        SecurityFactor: 0.6
112      (3,0):
113        Os: windows
114        Services: [ smtp, dns ]
115        Processes: [ Sendmail ]
116        SecurityFactor: 0.5
117      (4,0):
118        Os: linux
119        Services: [ http, https, dns, ftp ]
120        Processes: [ MySQL ]
121        SecurityFactor: 0.4
```

در شکل زیر گراف مربوط به Network را مشاهده می کنید.



Attack

این بخش از یک گراف تشکیل شده است. به این صورت که در هر Node اطلاعات زیر موجود است.

- 1- نام Exploit
- 2- نام آسیب پذیری
- 3- نام سیستم عامل مربوطه (در غیر این صورت None)
- 4- نام سرویس مربوطه (در غیر این صورت None)
- 5- نام Process مربوطه (در غیر این صورت None)
- 6- آدرس Host ای که مورد حمله قرار گرفته
- 7- احتمال موفقیت
- 8- Node بعدی در صورت موفقیت حمله (در صورت اتمام None)
- 9- Node بعدی در صورت شکست حمله (در صورت اتمام None)

10- حمله در چه مرحله ای قرار دارد از جمله (Data Initial Compromise, Exfiltration, Terminate Node)

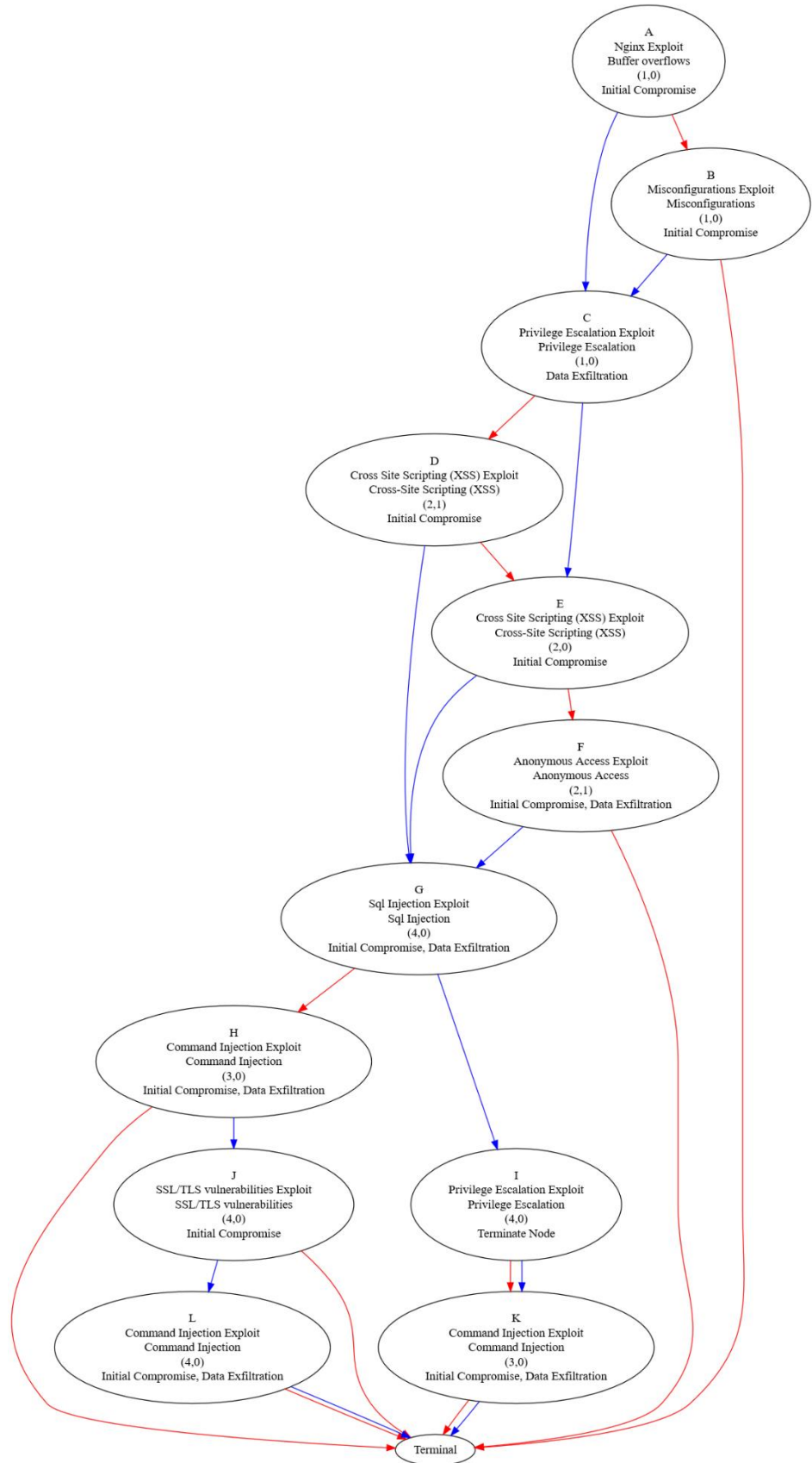
نرخ احتمال موفقیت را می توان با فرمول نسبت تعداد Exploit هایی که حمله کننده می داند به تمام Exploit هایی که برای آسیب پذیری های آن Host وجود دارد.

اما بهتر است که فرمول را در نظر نگیریم چرا که مهارت حمله کننده، حمله های روز صفر و ... را با فرمول در نظر نگرفته ایم.

مثالی از چند Node یک Attack را در شکل زیر می بینید.

در شکل بعد از آن مثالی از گراف حمله را مشاهده می کنید.

```
model1.yml x
123 Attack:
124   A:
125     ExploitName: Nginx Exploit
126     Vulnerability: Buffer overflows
127     Os: None
128     Service: None
129     Process: Nginx
130     SuccessRate: 0.8
131     Target: (1,0)
132     AttackStage: [Initial Compromise]
133     SuccessPath: C
134     FailurePath: B
135   B:
136     ExploitName: Misconfigurations Exploit
137     Vulnerability: misconfigurations
138     Os: None
139     Service: None
140     Process: Nginx
141     SuccessRate: 0.7
142     Target: (1,0)
143     AttackStage: [Initial Compromise]
144     SuccessPath: C
145     FailurePath: None
146   C:
147     ExploitName: Privilege Escalation Exploit
148     Vulnerability: privilege escalation
149     Os: linux
150     Service: None
151     Process: None
152     SuccessRate: 0.6
153     Target: (1,0)
154     AttackStage: [Data Exfiltration]
155     SuccessPath: E
156     FailurePath: D
157   D:
158     ExploitName: Cross-Site Scripting (XSS) Exploit
159     Vulnerability: Cross-Site Scripting (XSS)
160     Os: None
161     Service: http
162     Process: Apache
```



اینکه حمله ای در یک Node رخ می دهد یا خیر با توجه به فرمول زیر مشخص می شود.

احتمال موفقیت در آن Node ضربدر یک منهای Security Factor آن Host مربوطه

حال اگر برای مثال 10000 حمله را شبیه سازی کنیم با توجه به احتمالات حمله به مثال شکل

زیر می رسیم. * (در مورد تعداد حمله در بخش اصلاحات توضیحاتی آورده می شود)

model1.yml × AttackPath.txt ×

```
1  {'A", "B", "C", "D", "E", "F", "G", "H", "J", "L", "None:F"}': 5,
2  ['A", "B", "C", "D", "E", "F", "G", "H", "J", "L", "None:S"}': 1,
3  ['A", "B", "C", "D", "E", "F", "G", "H", "J", "None"}': 25,
4  ['A", "B", "C", "D", "E", "F", "G", "H", "None"}': 46,
5  ['A", "B", "C", "D", "E", "F", "G", "I", "K:F", "None:F"}': 26,
6  ['A", "B", "C", "D", "E", "F", "G", "I", "K:F", "None:S"}': 15,
7  ['A", "B", "C", "D", "E", "F", "G", "I", "K:S", "None:F"}': 20,
8  ['A", "B", "C", "D", "E", "F", "G", "I", "K:S", "None:S"}': 16,
9  ['A", "B", "C", "D", "E", "F", "None"}': 514,
10 ['A", "B", "C", "D", "E", "G", "H", "J", "L", "None:F"}': 1,
11 ['A", "B", "C", "D", "E", "G", "H", "J", "L", "None:S"}': 1,
12 ['A", "B", "C", "D", "E", "G", "H", "J", "None"}': 7,
13 ['A", "B", "C", "D", "E", "G", "H", "None"}': 16,
14 ['A", "B", "C", "D", "E", "G", "I", "K:F", "None:F"}': 11,
15 ['A", "B", "C", "D", "E", "G", "I", "K:F", "None:S"}': 7,
16 ['A", "B", "C", "D", "E", "G", "I", "K:S", "None:F"}': 6,
17 ['A", "B", "C", "D", "E", "G", "I", "K:S", "None:S"}': 3,
18 ['A", "B", "C", "D", "G", "H", "J", "L", "None:F"}': 6,
19 ['A", "B", "C", "D", "G", "H", "J", "L", "None:S"}': 2,
20 ['A", "B", "C", "D", "G", "H", "J", "None"}': 65,
21 ['A", "B", "C", "D", "G", "H", "None"}': 80,
22 ['A", "B", "C", "D", "G", "I", "K:F", "None:F"}': 39,
23 ['A", "B", "C", "D", "G", "I", "K:F", "None:S"}': 22,
24 ['A", "B", "C", "D", "G", "I", "K:S", "None:F"}': 40,
25 ['A", "B", "C", "D", "G", "I", "K:S", "None:S"}': 20,
26 ['A", "B", "C", "E", "F", "G", "H", "J", "None"}': 4,
27 ['A", "B", "C", "E", "F", "G", "H", "None"}': 9,
28 ['A", "B", "C", "E", "F", "G", "I", "K:F", "None:F"}': 8,
29 ['A", "B", "C", "E", "F", "G", "I", "K:F", "None:S"}': 1,
30 ['A", "B", "C", "E", "F", "G", "I", "K:S", "None:F"}': 4,
31 ['A", "B", "C", "E", "F", "G", "I", "K:S", "None:S"}': 3,
32 ['A", "B", "C", "E", "F", "None"}': 98,
33 ['A", "B", "C", "E", "G", "H", "J", "None"}': 1,
34 ['A", "B", "C", "E", "G", "H", "None"}': 1,
35 ['A", "B", "C", "E", "G", "I", "K:F", "None:F"}': 3,
36 ['A", "B", "C", "E", "G", "I", "K:S", "None:F"}': 1,
37 ['A", "B", "None"}': 7238,
38 ['A", "C", "D", "E", "F", "G", "H", "J", "L", "None:F"}': 9,
39 ['A", "C", "D", "E", "F", "G", "H", "J", "L", "None:S"}': 1,
40 ['A", "C", "D", "E", "F", "G", "H", "J", "None"}': 36,
41 ['A", "C", "D", "E", "F", "G", "H", "None"}': 83.
```

TODO Python Console Problems Terminal Services

به دلیل آنکه بتوان از روی Attack Path بتوان موفقیت آمیز بودن حمله در Node ها را تشخیص دهیم، زمانی که در صورت شکست و موفقیت در یک Node، Node بعدی یکسان باشد در اسم Node بعدی دو نقطه و F به معنی شکست و S به معنی پیروزی را اضافه کرده ایم.

کد

در کد برای شبیه سازی CSA به عددی با عنوان CSA Correctness نیاز داریم به این معنا که چند درصد مواقع پیشبینی قطعا درست است و چند درصد مواقع از روش دیگری استفاده شده است.

در کد از چهار CSA استفاده کرده ایم که اولی 0.5 و دومی 0.7 و سومی 0.9 Correctness دارند. برای چهارمین CSA از حالت رندوم استفاده کرده ایم به این معنا که برای هر پیشبینی یکبار این عدد به صورت تصادفی انتخاب می شود.

هر CSA در هر حمله دو پیشبینی را انجام می دهد. 1- هنگامی که حمله انجام شده است مشخص می کند که چه تغییراتی در وضعیت Host ها ایجاد شده است. 2- پیشبینی می کند که در صورت حمله بعدی چه تغییراتی در وضعیت Host ها ایجاد می شود.

به هر CSA چهار عدد را به عنوان احتمال می دهیم به صورتی که اولی احتمال رخ دادن حمله دومی احتمال رخ دادن CompleteCompromised و سومی احتمال رخ دادن DataLeakation و چهارمی احتمال Terminate شدن node (اگر از قبل CompleteCompromised شده باشد) را می دهد. که ما برای همه آن ها 0.5 را در نظر گرفته ایم.

در واقع CSA در هر پیشبینی ابتدا با استفاده از اعداد تصادفی مشخص می کند که باید جواب درست را برگرداند یا خیر. اگر عدد تصادفی کمتر از Correctness بود وضعیت درست شبکه را بر می گرداند. در غیر این صورت Host مربوطه را تشخیص داده و با احتمالات بالا وضعیت را بر می گرداند.

بعد از مشخص شدن هر پیشبینی وضعیت Business Importance فعلی شبکه را اعلام می کند.

در شکل زیر دو مثال از دیتاست که این اعداد برای هر چهار CSA آورده شده اند را مشاهده می کنید. * (در مورد فیلد های دیتاست شکل زیر در اصلاحات توضیحاتی ارائه می شود)

The first screenshot shows a data table with the following JSON structure:

```
{
  "_id": "ObjectId('66dd4ec44ae57c82f18ad72d')",
  "AttackPath": "[\"A\", \"B\", \"None\"]",
  "AttackPathNumber": 1,
  "FirstNode": "A",
  "SecondNode": "B",
  "RealBusinessFactor": 16,
  "FutureRealBusinessFactor": 16,
  "BusinessFactorCSA1": 16,
  "FutureBusinessFactorCSA1": 16,
  "BusinessFactorCSA2": 16,
  "FutureBusinessFactorCSA2": 16,
  "BusinessFactorCSA3": 16,
  "FutureBusinessFactorCSA3": 16,
  "BusinessFactorCSA4": 16,
  "FutureBusinessFactorCSA4": 16,
  "NodeNumber": 1
}
```

The second screenshot shows a filtered view of the data where FirstNode is 'C'. The JSON structure is:

```
{
  "_id": "ObjectId('66dd4ed24ae57c82f18b9fba')",
  "AttackPath": "[\"A\", \"C\", \"D\", \"G\", \"I\", \"K:S\", \"None:F\"]",
  "AttackPathNumber": 1,
  "FirstNode": "C",
  "SecondNode": "D",
  "RealBusinessFactor": 14.15,
  "FutureRealBusinessFactor": 13.41,
  "BusinessFactorCSA1": 14.15,
  "FutureBusinessFactorCSA1": 13.41,
  "BusinessFactorCSA2": 14.15,
  "FutureBusinessFactorCSA2": 13.41,
  "BusinessFactorCSA3": 14.15,
  "FutureBusinessFactorCSA3": 13.41,
  "BusinessFactorCSA4": 9.03,
  "FutureBusinessFactorCSA4": 13.41,
  "NodeNumber": 1
}
```

حال برای ارزیابی عملکرد یک CSA باید میانگین اختلاف عدد های پیشبینی شده را با اعداد واقعی به دست بیاوریم.

در شکل زیر این اعداد را می بینیم.

```
model1.yml x CSAResult.txt x
1 ([{'csa_number': 1,
2   'evaluate_future': 0.654498691410897,
3   'evaluate_now': 0.7475814928260519},
4   {'csa_number': 2,
5   'evaluate_future': 0.4042969307637402,
6   'evaluate_now': 0.47231178462034495},
7   {'csa_number': 3,
8   'evaluate_future': 0.13862669521770166,
9   'evaluate_now': 0.16109044010962437},
10  {'csa_number': 4,
11   'evaluate_future': 0.673604568165596,
12   'evaluate_now': 0.7617620506206674}],
```

اگر این اعداد را به درصد تبدیل کنیم داریم.

```
[{'csa_number': 1,
  'evaluate_future': 4.090616821318107,
  'evaluate_now': 4.672384330162824},
 {'csa_number': 2,
  'evaluate_future': 2.526855817273376,
  'evaluate_now': 2.9519486538771558},
 {'csa_number': 3,
  'evaluate_future': 0.8664168451106353,
  'evaluate_now': 1.0068152506851522},
 {'csa_number': 4,
  'evaluate_future': 4.210028551034975,
  'evaluate_now': 4.761012816379171}],
```

حال اگر زمان هایی را که اختلاف صفر است را در نظر نگیریم به اعداد زیر می رسم.

```
[{'csa_number': 1,
  'evaluate_future': 2.3866545202151657,
  'evaluate_now': 2.7210702969135077},
 {'csa_number': 2,
  'evaluate_future': 2.373931265716681,
  'evaluate_now': 2.7319563595673255},
 {'csa_number': 3,
  'evaluate_future': 2.3550848827809214,
  'evaluate_now': 2.764925290536801},
 {'csa_number': 4,
  'evaluate_future': 2.3992881355932205,
  'evaluate_now': 2.7459379358437936}],
```

اگر آن ها را به درصد تبدیل کنیم داریم.

```
[{'csa_number': 1,
  'evaluate_future': 14.916590751344785,
  'evaluate_now': 17.006689355709423},
 {'csa_number': 2,
  'evaluate_future': 14.837070410729256,
  'evaluate_now': 17.074727247295783},
 {'csa_number': 3,
  'evaluate_future': 14.719280517380758,
  'evaluate_now': 17.280783065855008},
 {'csa_number': 4,
  'evaluate_future': 14.99555084745763,
  'evaluate_now': 17.16211209902371}]]
```

اصلاحات

محاسبه اهمیت یک فرآیند

اگر فرآیند Process Data را در نظر بگیریم، این فرآیند در مسیری با ضریب اهمیت 5 قرار دارد و در این مسیر 4 فرآیند وجود دارد پس اهمیت این فرآیند برابر با $(5/4=1.25)$ است.

این فرآیند در ماموریت Process Data Correctly نیز وجود دارد که ضریب اهمیت آن برابر 4 است و در آن 3 فرآیند وجود دارد پس اهمیت این فرآیند در بخش ماموریت ها برابر با $(4/3=1.3)$ است.

ضریب اهمیت یک فرآیند از مجموع اهمیت آن فرآیند در بخش Workflow و بخش Mission بدست می آید که در اینجا برابر با $(1.25+1.3=2.55)$ است.

در شکل زیر محاسبه برای همه فرآیند ها را در کد مشاهده می کنید.

```
self.processes = (list: 5) [{ Importance: 3.08333333333333, Name: 'ReceiveRequest', RelatedResourcePool: 'NginxServer'}, {'Importance': 3.08333333333333, 'Name': 'ReceiveRequest', 'RelatedResourcePool': 'NginxServer'}
> 0 = {dict: 3} {'Importance': 3.08333333333333, 'Name': 'ReceiveRequest', 'RelatedResourcePool': 'NginxServer'}
> 1 = {dict: 3} {'Importance': 4.75, 'Name': 'Authenticate', 'RelatedResourcePool': 'Servers'}
> 2 = {dict: 3} {'Importance': 2.5, 'Name': 'EmailUser', 'RelatedResourcePool': 'SMTPServer'}
> 3 = {dict: 3} {'Importance': 2.58333333333333, 'Name': 'ProcessData', 'RelatedResourcePool': 'Servers'}
> 4 = {dict: 3} {'Importance': 3.08333333333333, 'Name': 'SendResponse', 'RelatedResourcePool': 'NginxServer'}
10 __len__ = {int} 5
> Protected Attributes

Set value F2
```

محاسبه اهمیت یک host

برای محاسبه میزان اهمیت یک host ابتدا باید میزان اهمیت subnet مربوطه را بدست آوریم سپس اهمیت بدست آمده را تقسیم بر تعداد host های درون subnet کنیم.

برای بدست آوردن اهمیت یک subnet از اهمیت فرآیند ها استفاده می کنیم بدین صورت که می دانیم هر فرآیند به کدام ResourcePool مربوط است و هر ResourcePool مربوط به کدام subnet است. اگر ResourcePool وابستگی به ResourcePool های دیگر نداشته باشد همان اهمیت فرآیند به اهمیت subnet اضافه می شود در غیر این صورت میزان اهمیت بین این

ResourcePool (و به طبع آن subnet مربوطه) و ResourcePool های وابسته تقسیم می شود.

در شکل زیر اعداد اهمیت هر subnet محاسبه شده است.

```
▼ subnet_data = {list: 4} [{"Importance": 6.166666666666666, "Number": 1}, {"Importance": 4.916666666666666, "Number": 2}, {"Importance": 1.25, "Number": 3}, {"Importance": 3.666666666666665, "Number": 4}]
  0 = {dict: 2} {"Importance": 6.166666666666666, "Number": 1}
  1 = {dict: 2} {"Importance": 4.916666666666666, "Number": 2}
  2 = {dict: 2} {"Importance": 1.25, "Number": 3}
  3 = {dict: 2} {"Importance": 3.666666666666665, "Number": 4}
  __len__ = {int} 4
  Protected Attributes
Set value F2
```

در شکل زیر اعداد اهمیت هر host محاسبه شده است.

```
▼ self.hosts_configuration = {list: 5} [{"Address": "(1,0)", "Importance": 6.166666666666666, "Os": "linux", "Processes": ["Nginx"], "SecurityFactor": 0.8, "Services": ["http", "https"]}, {"Address": "(2,0)", "Importance": 2.458333333333333, "Os": "linux", "Processes": ["Apache"], "SecurityFactor": 0.9, "Services": ["http", "https"]}, {"Address": "(2,1)", "Importance": 2.458333333333333, "Os": "linux", "Processes": ["Apache"], "SecurityFactor": 0.6, "Services": ["http", "https"]}, {"Address": "(3,0)", "Importance": 1.25, "Os": "windows", "Processes": ["Sendmail"], "SecurityFactor": 0.5, "Services": ["smtp", "dns"]}, {"Address": "(4,0)", "Importance": 3.666666666666665, "Os": "linux", "Processes": ["MySQL"], "SecurityFactor": 0.4, "Services": ["http", "https"]}
  0 = {dict: 6} {"Address": "(1,0)", "Importance": 6.166666666666666, "Os": "linux", "Processes": ["Nginx"], "SecurityFactor": 0.8, "Services": ["http", "https"]}
  1 = {dict: 6} {"Address": "(2,0)", "Importance": 2.458333333333333, "Os": "linux", "Processes": ["Apache"], "SecurityFactor": 0.9, "Services": ["http", "https"]}
  2 = {dict: 6} {"Address": "(2,1)", "Importance": 2.458333333333333, "Os": "linux", "Processes": ["Apache"], "SecurityFactor": 0.6, "Services": ["http", "https"]}
  3 = {dict: 6} {"Address": "(3,0)", "Importance": 1.25, "Os": "windows", "Processes": ["Sendmail"], "SecurityFactor": 0.5, "Services": ["smtp", "dns"]}
  4 = {dict: 6} {"Address": "(4,0)", "Importance": 3.666666666666665, "Os": "linux", "Processes": ["MySQL"], "SecurityFactor": 0.4, "Services": ["http", "https"]}
  __len__ = {int} 5
  Protected Attributes
Set value F2
```

بدست آوردن تعداد شبیه ساز حمله

به جای استفاده از عدد ثابت ده هزار تا می آیم و احتمال رخ دادن موفقیت آمیزترین مسیر حمله را به دست می آوریم. در این مدل مسیر حمله زیر است

["A", "C", "E", "G", "I", "K:S", "None:S"]

اگر احتمال موفقیت ها را در هم ضرب کنیم، احتمال رخ دادن آن مسیر حمله به دست می آید.

$$(0.8 * 0.2) * (0.6 * 0.2) * (0.7 * 0.1) * (0.7 * 0.6) * (0.7 * 0.6) * (0.8 * 0.5) = 0.00009483264$$

اگر عدد بدست آمده را ضربدر X کنیم و مقدار برابر با 1 شود آنگاه تعداد لازم را به دست می آوریم.

عدد بدست آمده را با صدگان آن رند می کنیم. یعنی 10,544.89256019868 برابر با 10500 می شود.

دیتاست

برای آن که وضعیت شبکه را در هر داده داشته باشیم فیلدی به نام state را اضافه کردیم که در آن وضعیت هر کدام از host ها آورده شده است.

csa_evaluating.csa_evaluating

65.4k 1
DOCUMENTS INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter ⓘ ⓘ Type a query: { field: 'value' }

Reset Find ⓘ More Options ▶

ADD DATA EXPORT COLLECTION

1 - 20 of 65388

```
{
  "_id": ObjectId("66f933c0ecf564b040014598"),
  "AttackPath": ["A", "B", "None"],
  "AttackPathNumber": 1,
  "FirstNode": "A",
  "SecondNode": "B",
  "RealBusinessFactor": 16,
  "State": Array
  * 0: Object
    Address: "(1,0)"
    attemptedAttack: true
    IsCompromised: false
    IsCompromisedCompletely: false
    IsTerminated: false
    IsDataLeaked: false
    Importance: 6.166666666666666
  * 1: Object
    Address: "(2,0)"
    attemptedAttack: false
    IsCompromised: false
    IsCompromisedCompletely: false
    IsTerminated: false
    IsDataLeaked: false
    Importance: 2.458333333333333
  * 2: Object
  * 3: Object
  * 4: Object
  FutureRealBusinessFactor: 16
  BusinessFactorCSA1: 14.15
  FutureBusinessFactorCSA1: 12.3
  BusinessFactorCSA2: 12.3
  FutureBusinessFactorCSA2: 16
  BusinessFactorCSA3: 16
  FutureBusinessFactorCSA3: 16
  BusinessFactorCSA4: 16
  FutureBusinessFactorCSA4: 16
  ModelNumber: 1
}
```

در هنگام تمام شدن باید فیلد های مربوط به CSA را حذف کنیم چرا که آن ها برای مقایسه استفاده شده اند و نیازی به حضور در دیتاست ندارند.