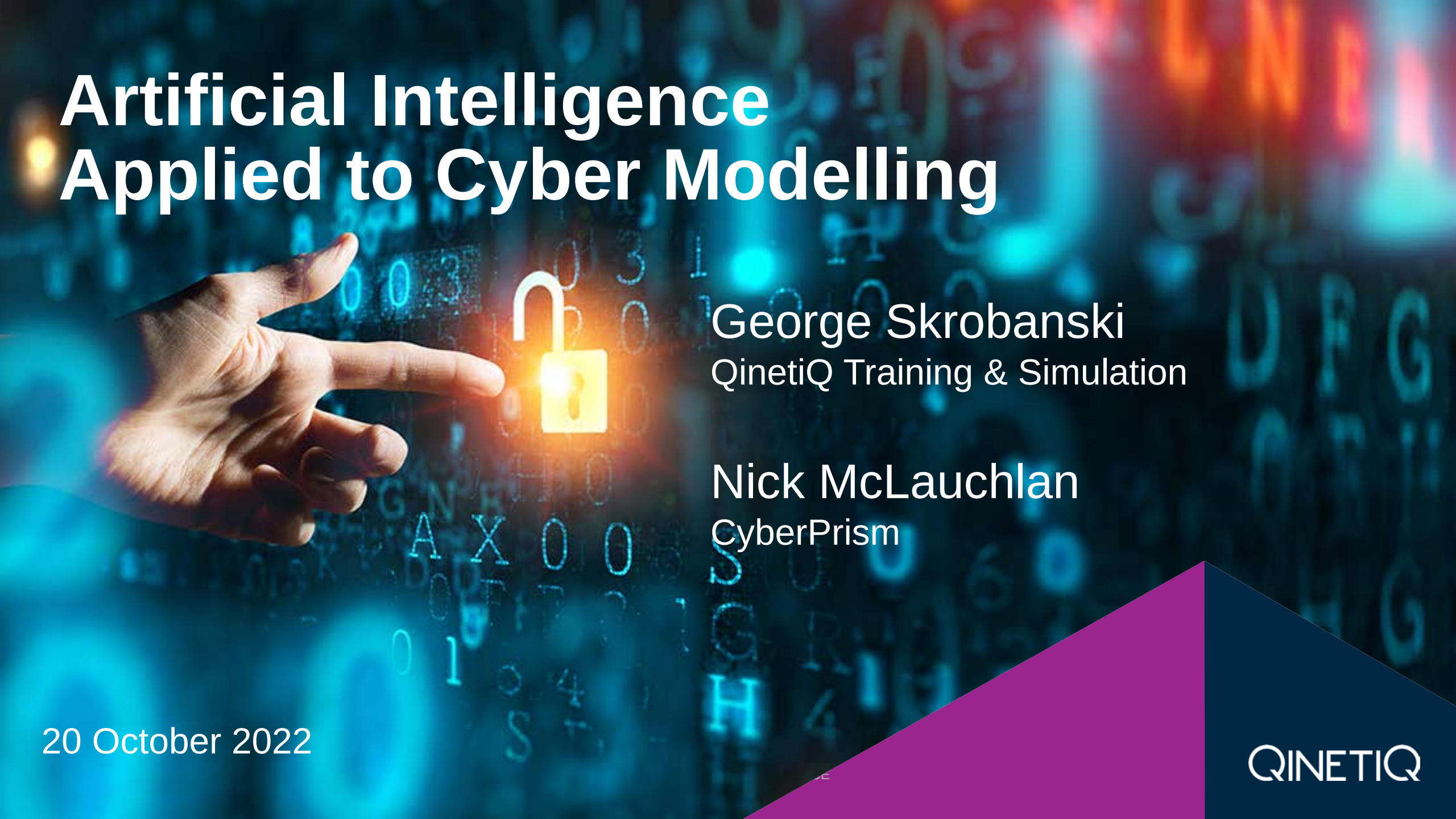
Artificial Intelligence Applied to Cyber Modelling



George Skrobanski
QinetiQ Training & Simulation

Nick McLauchlan
CyberPrism

20 October 2022

QINETIQ

Artificial Intelligence Applied to Cyber Modelling

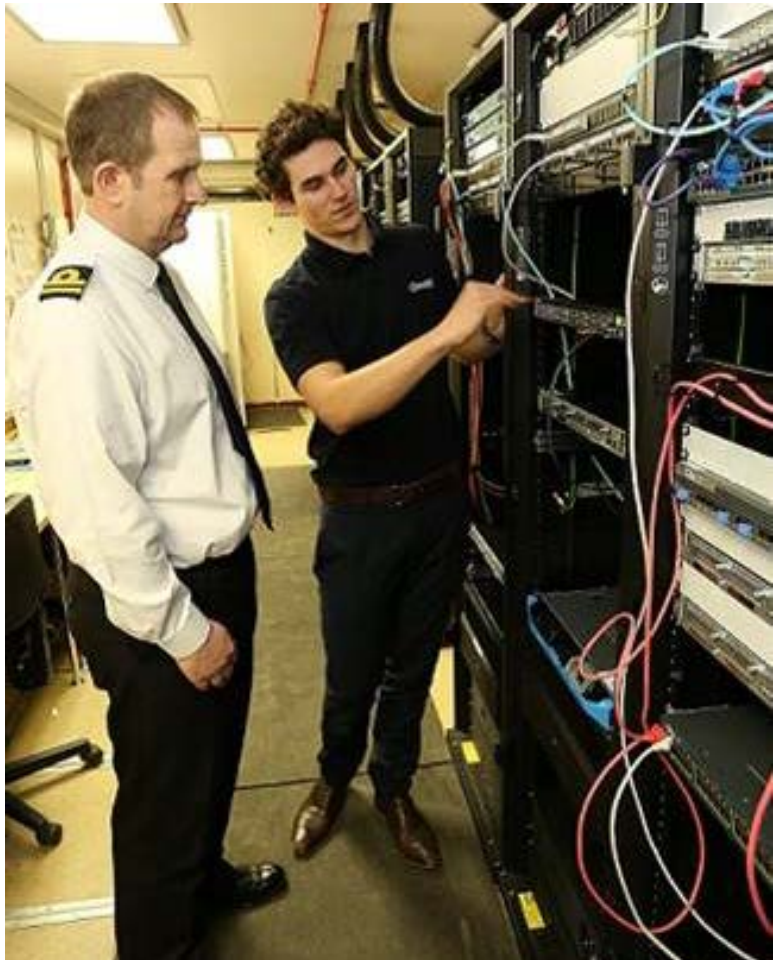
QINETIQ

1 Introduction

QINETIQ

1 Introduction (1 of 4)

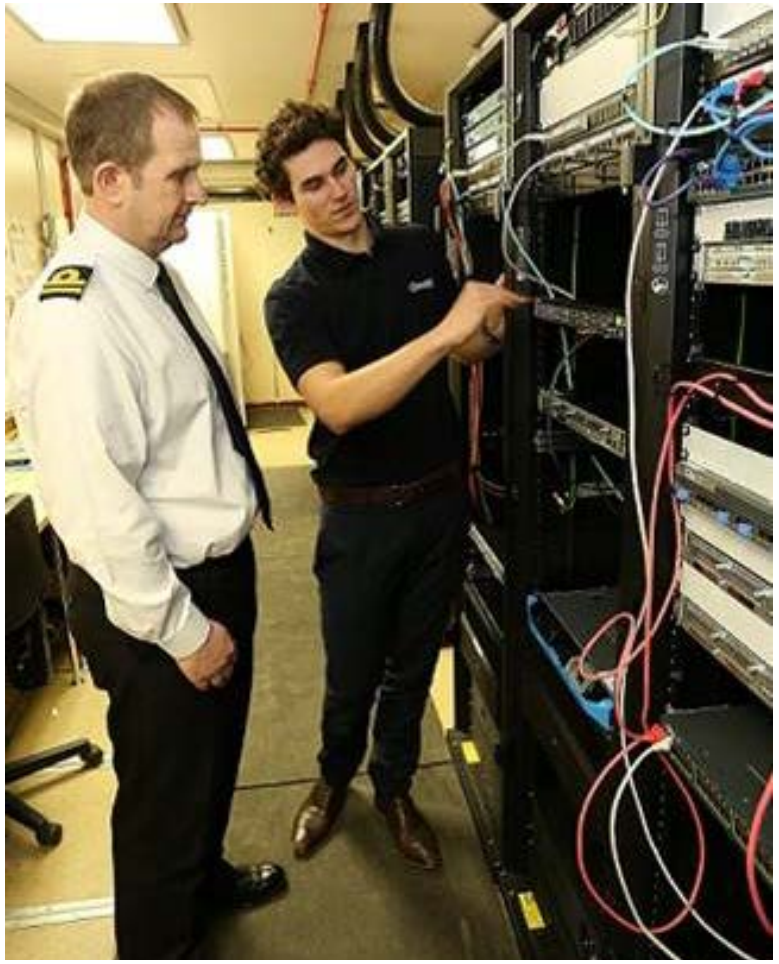
- Computer networks in the military domain, and everywhere, are increasingly subject to cyber attacks.
- It is important to be able to quickly detect vulnerabilities to such attacks and select suitable responses.
- There is widespread interest in the development of automated cyber defence processes using Artificial Intelligence (AI) technology.
- For example, in February 2020 Microsoft Research established a cyberattack simulator called CyberBattleSim which uses reinforcement learning (RL) algorithms via the open source AI Gym toolkit.
- A search on Google will quickly find half a dozen similar RL-based cyberattack simulators.



QINETIQ

1 Introduction (2 of 4)

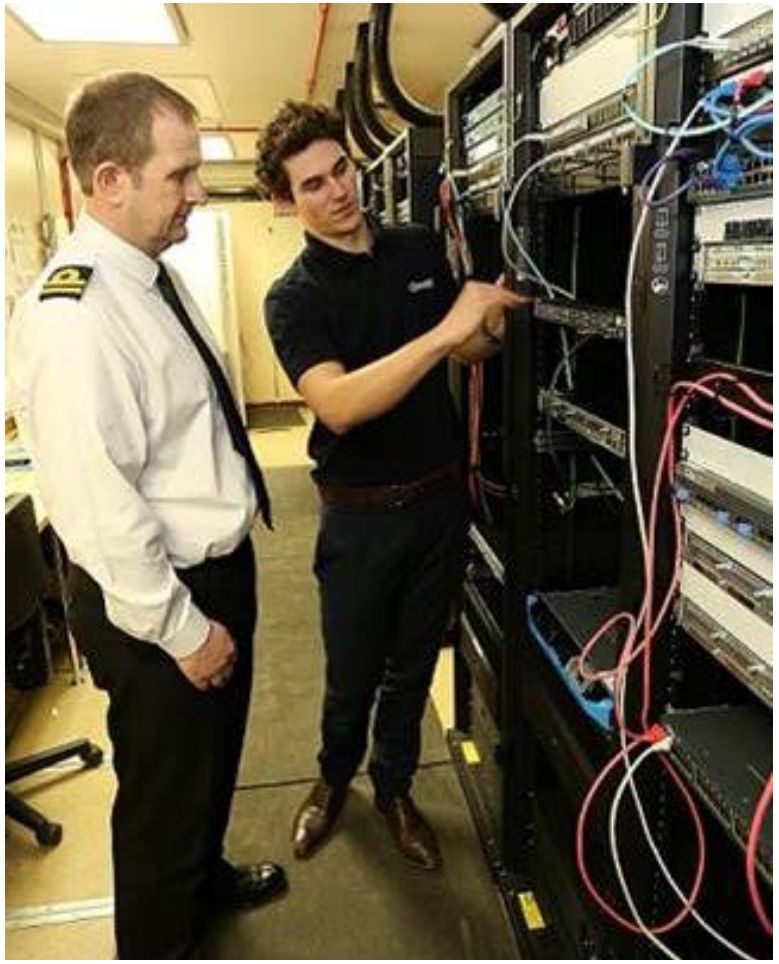
- By contrast, we have developed an automated AI cyber simulation that employs statistical forward planning (SFP) rather than RL.
- SFP algorithms are a family of robust stochastic AI techniques that use a statistical model to simulate possible future states.
- They operate without training and are for this reason much faster than learning-based methods.



QINETIQ

1 Introduction (3 of 4)

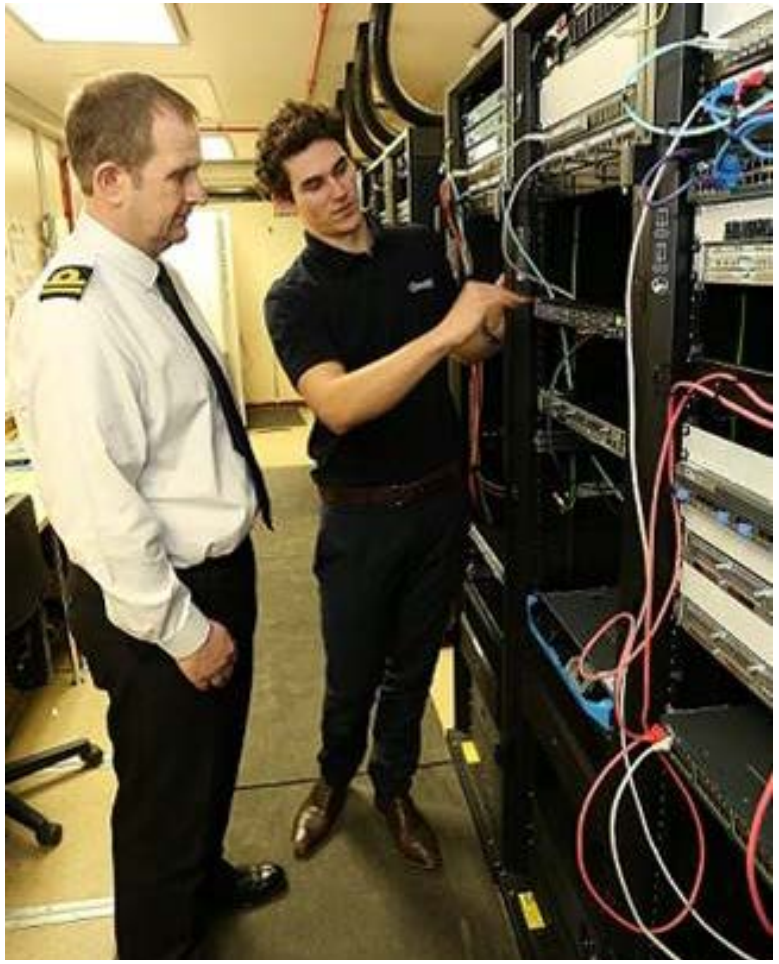
- The AI software used in this project is an extension of the Mission Planner AI tool developed by QinetiQ Training and Simulation.
- Originally, Mission Planner was applied to generate and analyse orders in land-based war games.
- Since then Mission Planner has also been successfully used in such contexts as antisubmarine warfare.
- The AI engine within Mission Planner, which by design knows nothing about the particular application, works only on the basis of the reward or utility of a particular sequence of orders.
- This gives Mission Planner considerable generality.



QINETIQ

1 Introduction (4 of 4)

- The AI engine within Mission Planner uses simulated annealing to select an optimal script.
- The scripts generated by the AI engine have no meaning within the engine itself.
- They are passed to an application specific component within Mission Planner called the Decoder.
- This computes a numerical value, called the reward or utility or value, which is passed back to the AI engine.
- The optimization algorithm proceeds iteratively, generating a sequence of scripts whose values converge gradually (but usually not monotonically) to an optimum.



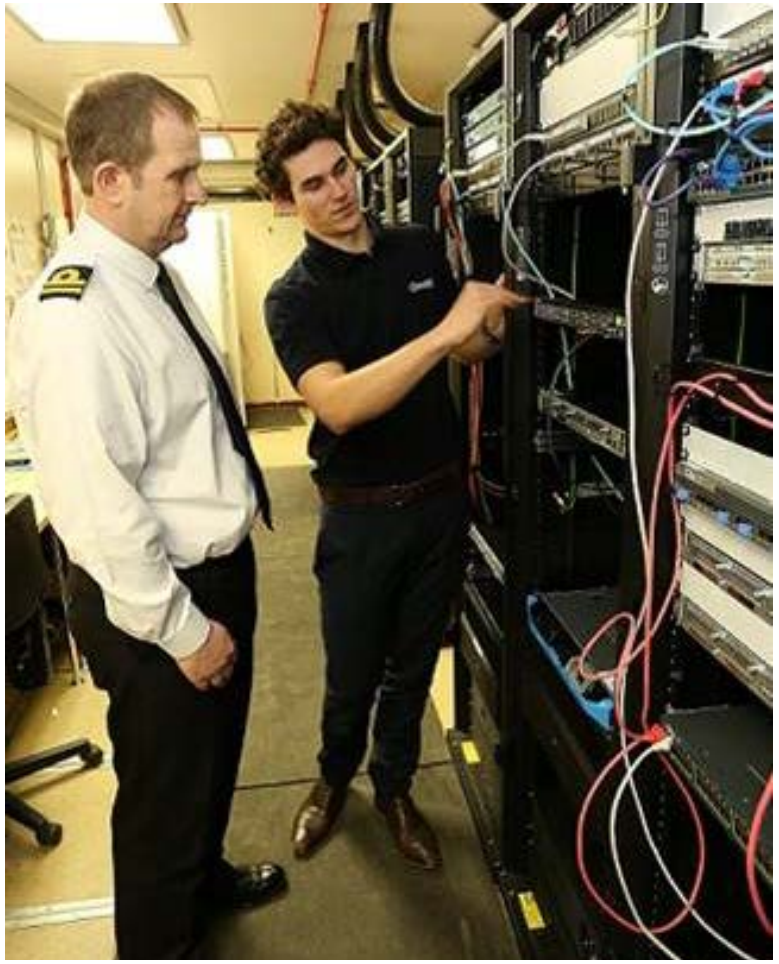
QINETIQ

2 Cyber Attack Structure

QINETIQ

2 Cyber Attack Structure (1 of 4)

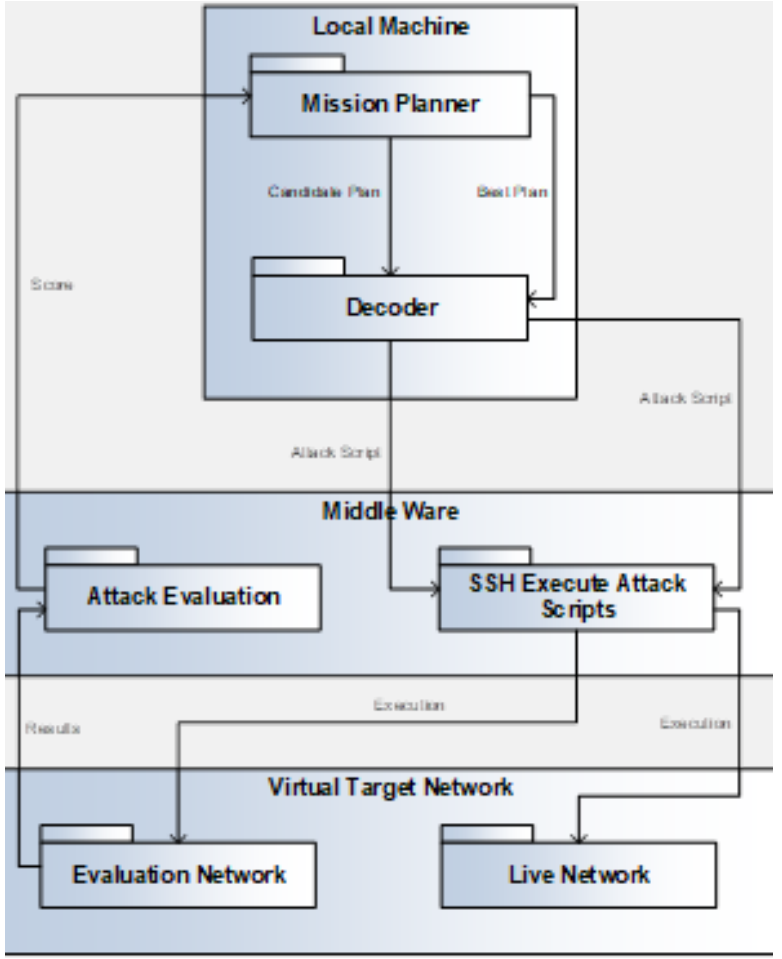
- A virtual target network containing several known vulnerabilities was setup on a remote sever.
- Example vulnerabilities might be that some of the machines might have older versions of operating systems that are missing important security patches.
- The target network communicates with the Mission Planner Decoder via a Middle Ware component.



QINETIQ

2 Cyber Attack Structure (2 of 4)

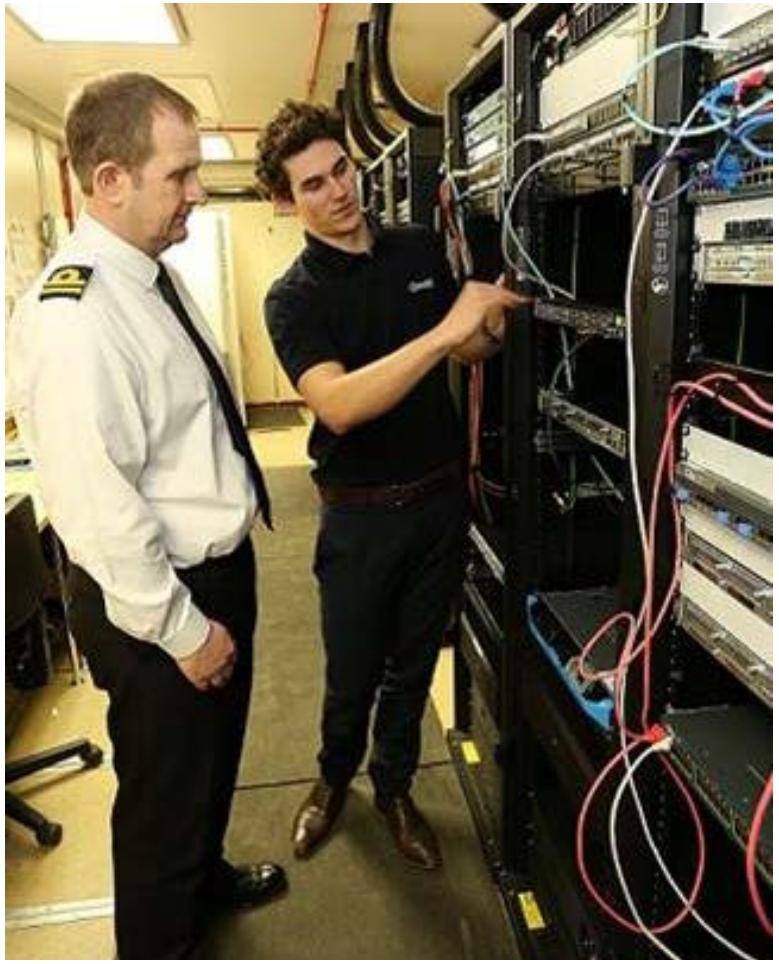
- The intended structure is shown in the diagram at right.
- The Decoder passes an attack script to the Middle Ware component.
- The Middle Ware component executes the attack script on the evaluation network.
- The results of the attack on the evaluation network are returned to the Middle Ware component which assesses the success of the attack and computes a resultant numerical score.
- This numerical score is passed back to the Decoder.



QINETIQ

2 Cyber Attack Structure (3 of 4)

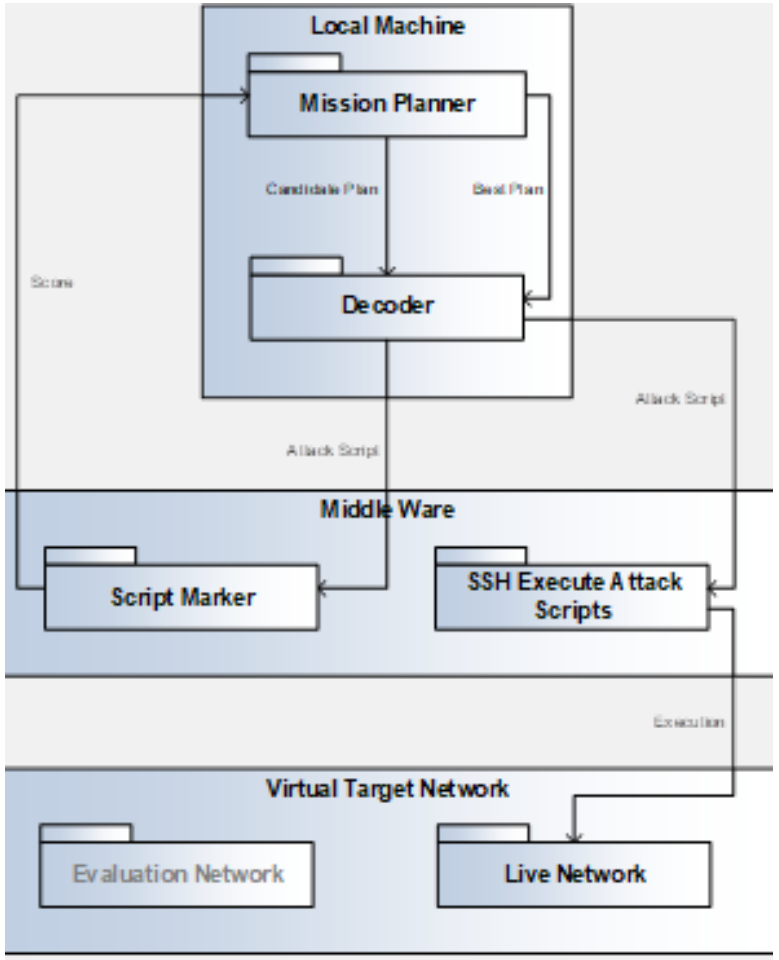
- The project initially considered using a Mininet based target environment, including Containernet and libvirt support, to allow for the use of docker containers and virtual machines.
- This would have allowed the rapid generation and testing of attack scripts on the evaluation network.
- Because the libvirt support for Containernet was experimental and had not been updated for 5 years, it was not possible to implement the target network using Mininet.



QINETIQ

2 Cyber Attack Structure (4 of 4)

- For this reason, the calls to the evaluation network were removed and replaced by a script marker that assesses the similarity of an AI generated script compared to a given manually generated one.
- The implemented cyber attack structure is shown in the diagram at right.



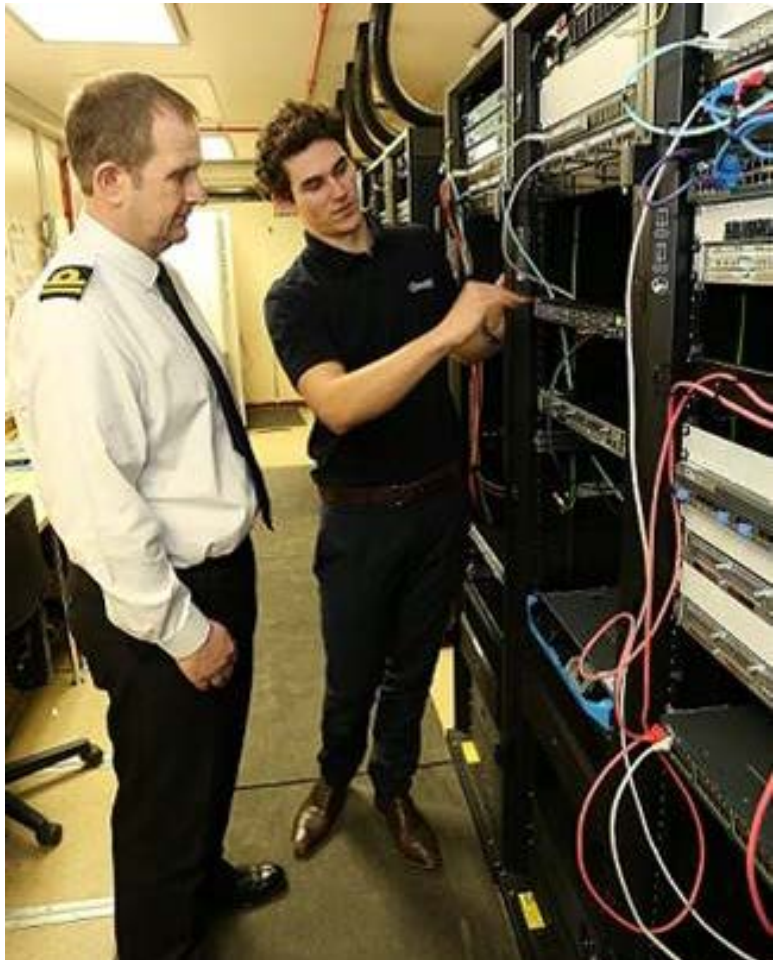
QINETIQ

3 The Script Marker

QINETIQ

3 The Script Marker (1 of 3)

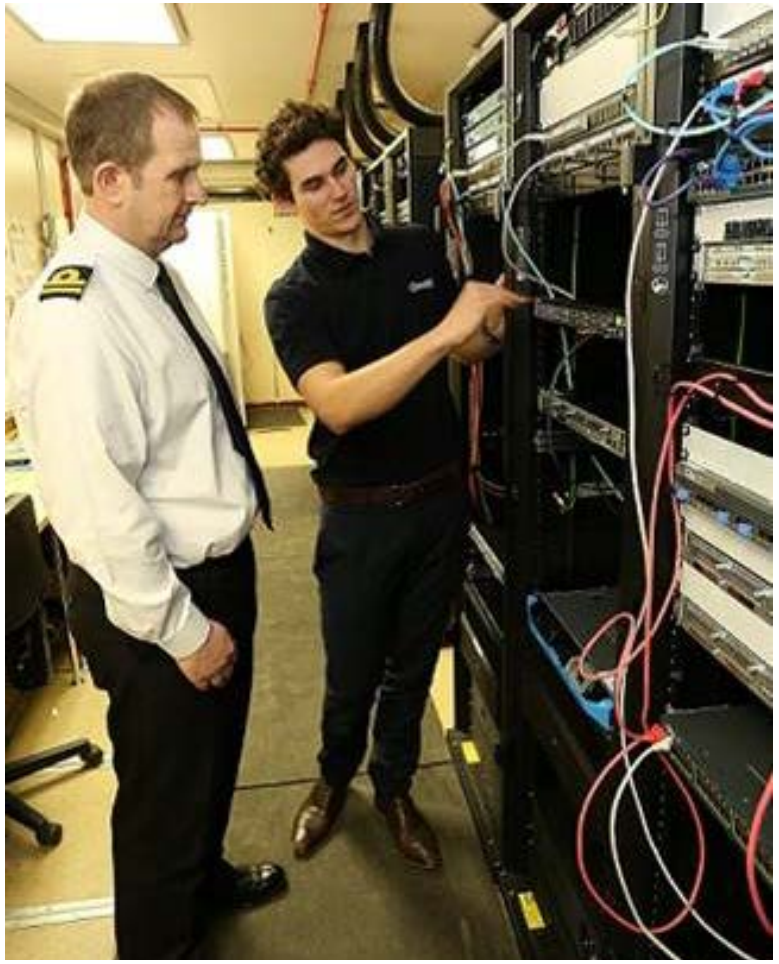
- Mission Planner scripts consist of trees, where each node in the tree is either:
  - an order node; or
  - an input node containing a numerical parameter for use with an order node.
- An input node is always a child of an order node, and the input node's parameter value is applied to the parent order node.
- An input node does not have any child nodes.
- Order nodes must always have at least one child node.
- Child nodes of an order node can be themselves either order nodes or input nodes.
- The order nodes can only be one of a finite number of possible types.



QINETIQ

3 The Script Marker (2 of 3)

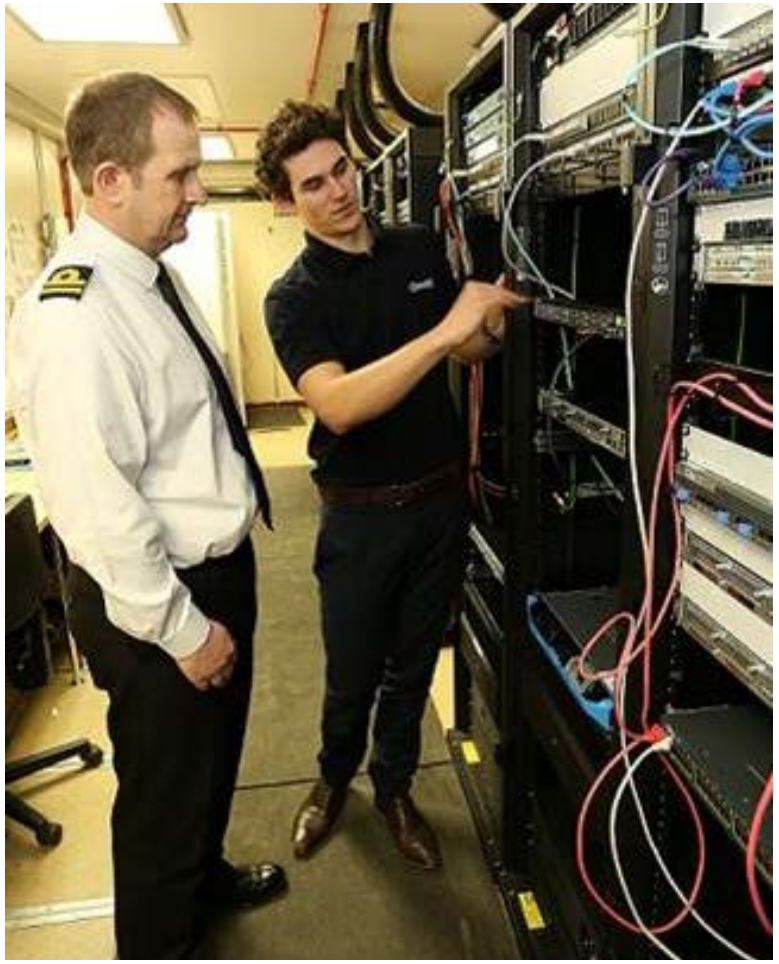
- For a given pair of order nodes of the same type, the script maker can compute a similarity value as follows:
  1. Initialize the value to zero.
  2. Add one (to the value) if both nodes do not have a parent.
  3. Add one if both nodes *do* have a parent and *both* parent nodes are of the same type.
  4. Iterate through each order node pair's input nodes, comparing the first entries in each list, the second entries, and so on, ending when one of the lists is exhausted; add one to the similarity value for every pair of input nodes that has the same value, and an additional one for every pair that is of the same type.
  5. Normalize the similarity value by dividing it by the number of child nodes of the parent node, using whichever parent node has the smallest number of child nodes.



QINETIQ

3 The Script Marker (3 of 3)

- The script marker computes an overall similarity score between two scripts A and B as follows:
  1. Initialize the score to zero.
  2. For each order node in script A, find the order node of the same type in script B that has largest similarity value, and add this maximum similarity value to the score.
  3. Normalize the score by diving it by the number of order nodes in the script, using whichever script has the largest number of order nodes.
  - This algorithm returns a value between zero and one inclusive.
  - It will return one if all the order node types and input value match exactly.
  - It will return zero if there are no matches.

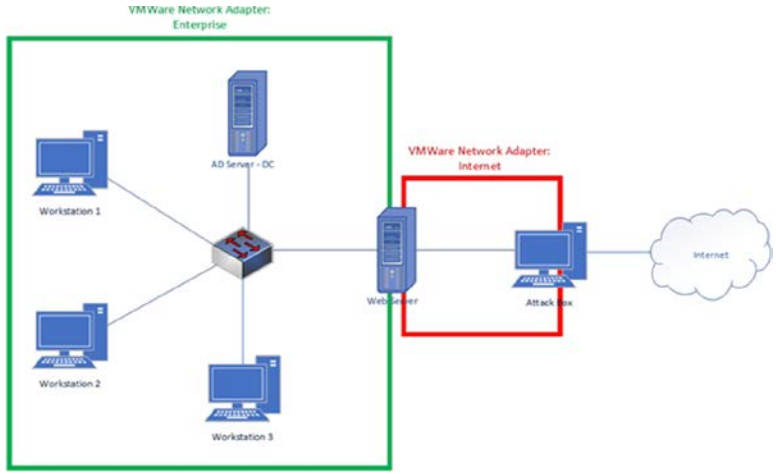


QINETIQ

4 The Live Network

4 The Live Network (1 of 3)

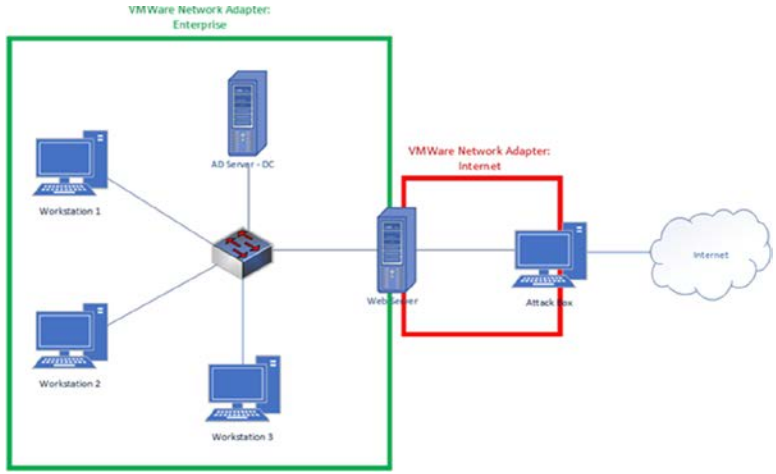
- The diagram shows the live network used to test an AI attack in which a dual-homed, domain connected webserver is compromised following the 7 steps of the Lockheed Cyber Kill Chain.
- On successful compromise of the webserver the AI repeats the process of reconnaissance, weaponization, delivery etc. to identify vulnerabilities in the domain and move its offensive tooling onto the compromised server, using this to capture the NTLM hash of a domain user.
- The AI copies the NTLM hash to the Attack Box it is using for initial access, where it uses John the Ripper to identify the username and password of the domain user.



QINETIQ

4 The Live Network (2 of 3)

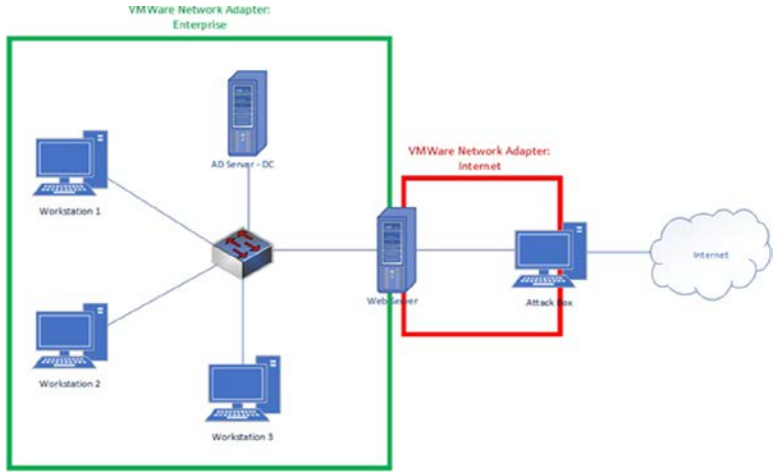
- In this instance the user account harvested is that of a Domain Administrator, with the AI now using those credentials to connect to the ADServer and create its own account on the domain.
- The green square on the left hand side contains the virtual network that is attacked.
- The red square in the middle of the figure contains the attack box which is the computer that was physically the source of the attack in this test.



QINETIQ

4 The Live Network (3 of 3)

- Due to the decentralised nature of the project as a result of Covid restrictions, the live network to be compromised and the computer hosting Mission Planner were geographically separated, necessitating the use of a ZeroTier Software Defined Network to provide secure communications.
- Initially the attack script was developed by hand on the attack box, and manually tested by observing the result of attacks on the webserver.
- Once a satisfactory attack script had been developed, it was incorporated into the script marker and the Mission Planner AI was used to generate an optimal attack script.
- This was then passed via the internet to the attack box, from which it was launched on the target network.



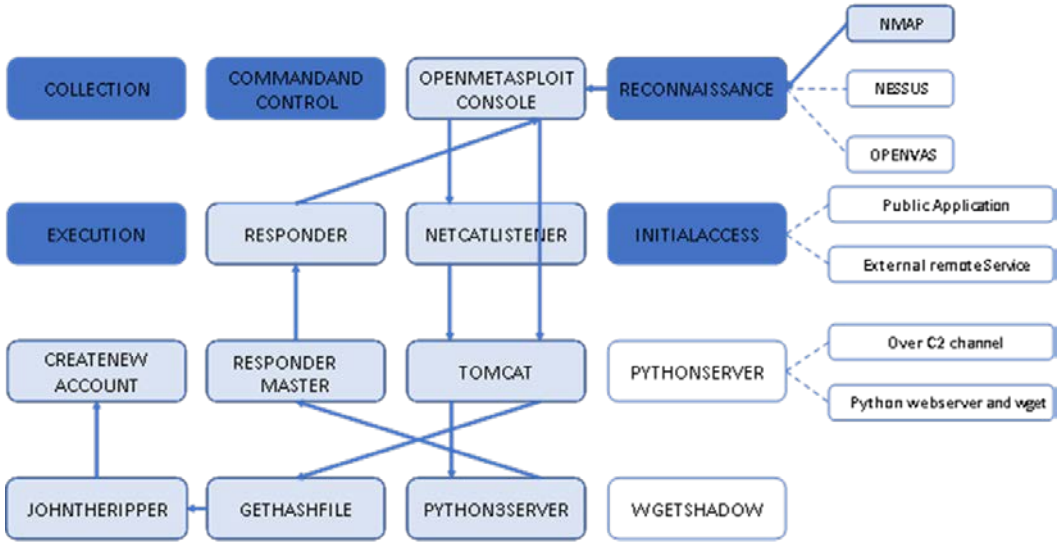
QINETIQ

5 AI Attack Syntax

QINETIQ

5 AI Attack Syntax (1 of 3)

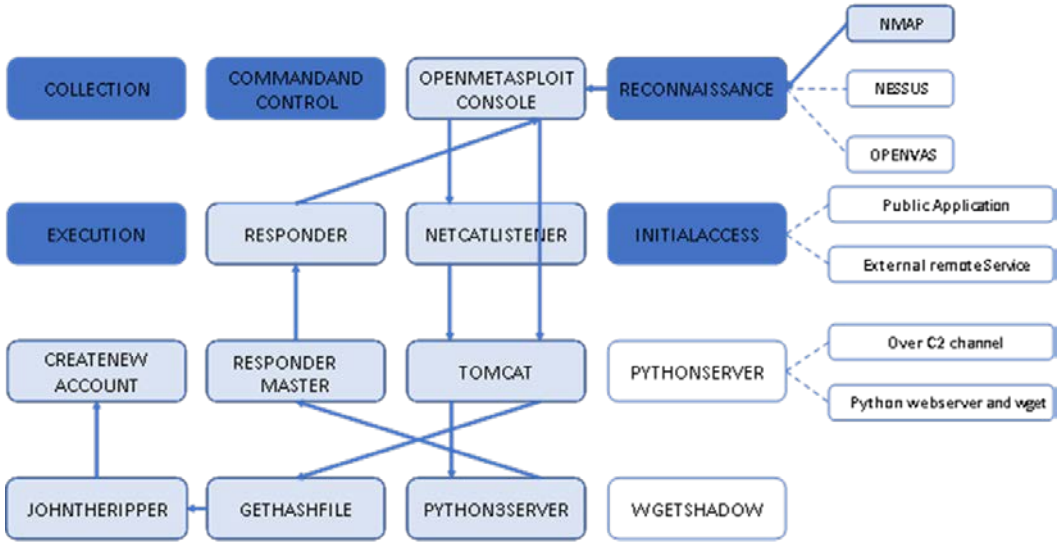
- The diagram shows the attack script components required for domain take over.
- Tactics are shaded in dark blue, techniques are shaded in light blue.
- Also shown are some additional components that are not required for the current attack.



QINETIQ

5 AI Attack Syntax (2 of 3)

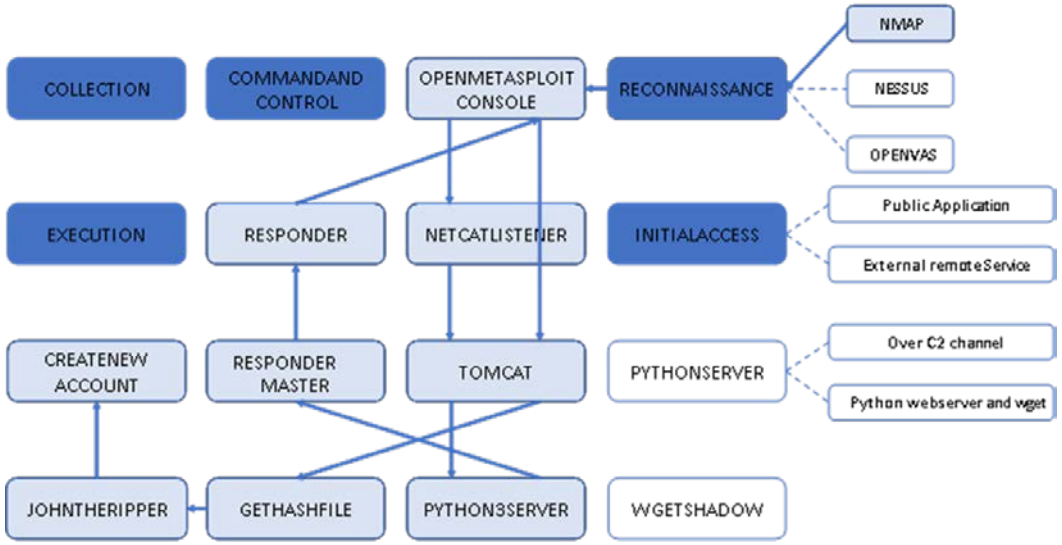
- In the diagram, the non required components come from a data exfiltration attack.
- Note also that PYTHONSERVER and PYTHON3SERVER are different instances; PYTHONSERVER is used for webserver and wget, and PYTHON3SERV must be python3.
- The project based the attack components on the tactics and techniques listed in MITRE ATT&CK®, a globally-accessible knowledge base of cyber tactics and techniques based on real-world observations.
- These components are provided to Mission Planner which must then assemble them in the correct order to achieve the desired outcome.



QINETIQ

5 AI Attack Syntax (3 of 3)

- Some components have a number of options.
- For example INITIALACCESS could be either Public Application, or External remote service.
- RECONNAISSANCE is more complex, in that it could be one of NMAP, NESSUS or OPENVAS and NMAP RECONNAISSANCE could be any of the available options, ranging from stealthy to aggressive.
- These component options are indicated by dashed lines in the diagram.
- The solid arrows indicate the correct sequence of script components required for the domain take-over attack.



QINETIQ

6 AI Performance

QINETIQ

6 AI Performance (1 of 3)

- Each generation considered 15,200 attack scripts
- Best and Average Score are the best and average similarity scores in the current generation, expressed as percentages.
- Largest is the number of components in the longest script in the current generation.
- Average Size is the average number of components in each script in the current generation.
- Cumulative Time is the elapsed time in seconds since the start of the optimization process.
- Fraction replaced is the fraction of candidate scripts accepted by the simulated annealing algorithm in the current generation.

Generation	Best Score	Average Score	Largest	Average Size	Time (d.hh:mm:ss)	Fraction Replaced
0	66.667	30.578	51	13.705	0.00:00:01.59	0.70527
1	72.222	31.733	79	14.931	0.00:00:03.40	0.69357
2	74.779	30.1	178	21.131	0.00:00:05.79	0.70196
3	74.779	33.219	66	14.805	0.00:00:07.44	0.6821
4	74.779	34.181	54	14.533	0.00:00:09.15	0.65849
5	74.779	34.214	55	15.028	0.00:00:11.27	0.66099
6	74.9	37.34	40	12.779	0.00:00:13.19	0.62636
7	74.9	39.803	34	13.049	0.00:00:14.73	0.60488
8	80.455	41.154	29	12.591	0.00:00:16.61	0.58879
9	80.556	43.014	33	12.573	0.00:00:18.46	0.58035
10	80.556	45.291	31	12.333	0.00:00:20.13	0.55896
11	84.363	47.624	37	12.738	0.00:00:21.94	0.54465
12	84.363	50.407	22	12.229	0.00:00:23.51	0.52152
13	86.764	54.748	19	12.019	0.00:00:25.05	0.47969
14	93.994	56.303	18	12.079	0.00:00:26.46	0.45686
15	93.994	61.159	19	11.917	0.00:00:28.26	0.39016
16	93.994	63.515	17	11.925	0.00:00:30.14	0.32042
17	99.121	70.429	15	11.919	0.00:00:31.71	0.20595
18	99.674	75.678	15	11.899	0.00:00:33.56	0.098688
19	99.992	88.921	14	11.994	0.00:00:35.32	0.018495
20	99.999	86.567	14	11.973	0.00:00:37.17	0.024843

QINETIQ

6 AI Performance (2 of 3)

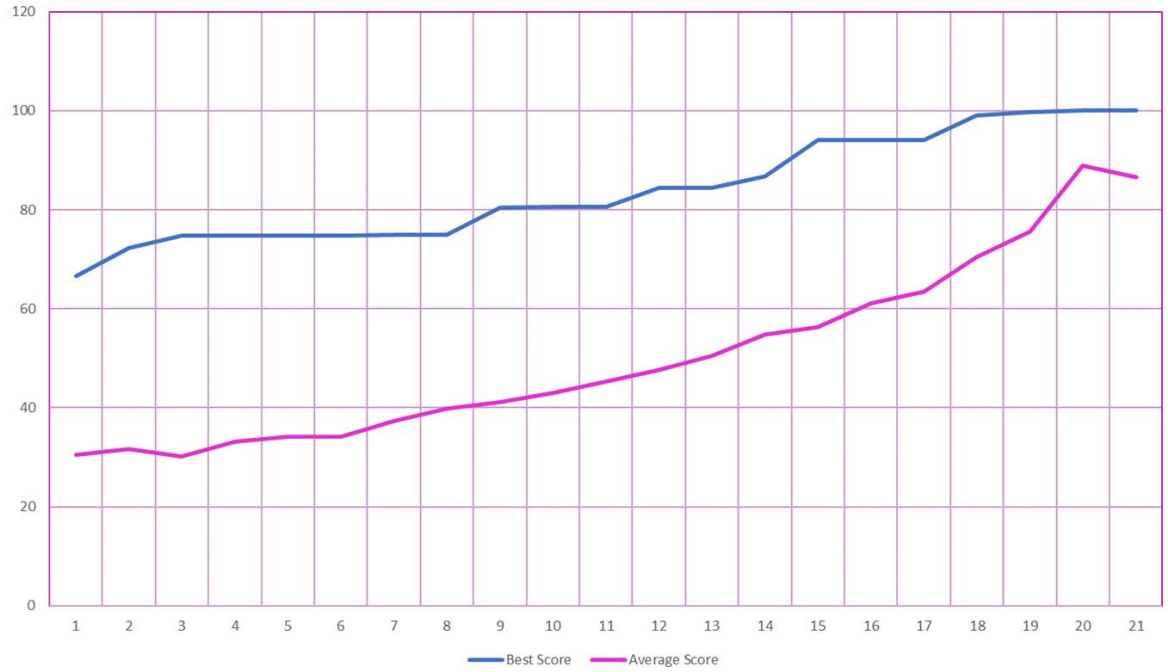
- The optimizer achieved a best score of 93.994% at generation 14.
- This script differed from the target script by only step, although it still resulted in a failed attack.
- A near perfect solution is obtained at generation 20, resulting in a successful attack.
- The total elapsed run time was 37 seconds.

Generation	Best Score	Average Score	Largest	Average Size	Time (d.hh:mm:ss)	Fraction Replaced
0	66.667	30.578	51	13.705	0.00:00:01.59	0.70527
1	72.222	31.733	79	14.931	0.00:00:03.40	0.69357
2	74.779	30.1	178	21.131	0.00:00:05.79	0.70196
3	74.779	33.219	66	14.805	0.00:00:07.44	0.6821
4	74.779	34.181	54	14.533	0.00:00:09.15	0.65849
5	74.779	34.214	55	15.028	0.00:00:11.27	0.66099
6	74.9	37.34	40	12.779	0.00:00:13.19	0.62636
7	74.9	39.803	34	13.049	0.00:00:14.73	0.60488
8	80.455	41.154	29	12.591	0.00:00:16.61	0.58879
9	80.556	43.014	33	12.573	0.00:00:18.46	0.58035
10	80.556	45.291	31	12.333	0.00:00:20.13	0.55896
11	84.363	47.624	37	12.738	0.00:00:21.94	0.54465
12	84.363	50.407	22	12.229	0.00:00:23.51	0.52152
13	86.764	54.748	19	12.019	0.00:00:25.05	0.47969
14	93.994	56.303	18	12.079	0.00:00:26.46	0.45686
15	93.994	61.159	19	11.917	0.00:00:28.26	0.39016
16	93.994	63.515	17	11.925	0.00:00:30.14	0.32042
17	99.121	70.429	15	11.919	0.00:00:31.71	0.20595
18	99.674	75.678	15	11.899	0.00:00:33.56	0.098688
19	99.992	88.921	14	11.994	0.00:00:35.32	0.018495
20	99.999	86.567	14	11.973	0.00:00:37.17	0.024843

QINETIQ

6 AI Performance (3 of 3)

- The diagram shows the improvement in percentage score achieved in each generation.
- The best score gradually increases to 100%.
- The average score, of course, is less good than the best score; however the difference diminishes as the optimum is approached.
- This is typical behaviour of the simulated annealing algorithm.

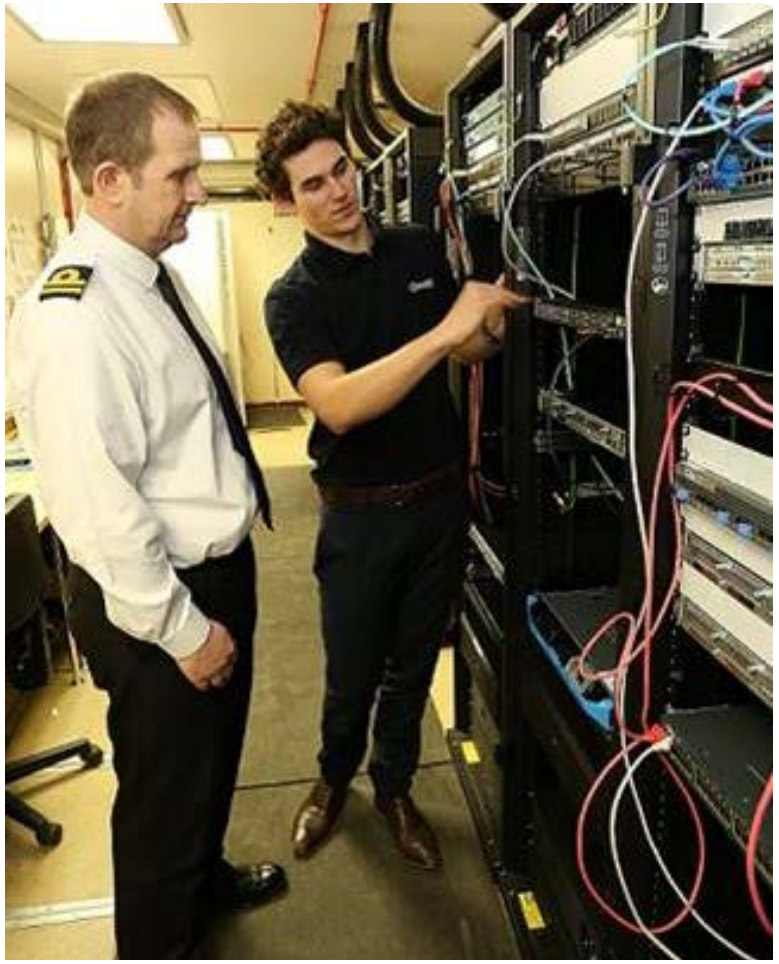


QINETIQ

7 Conclusions

7 Conclusions (1 of 3)

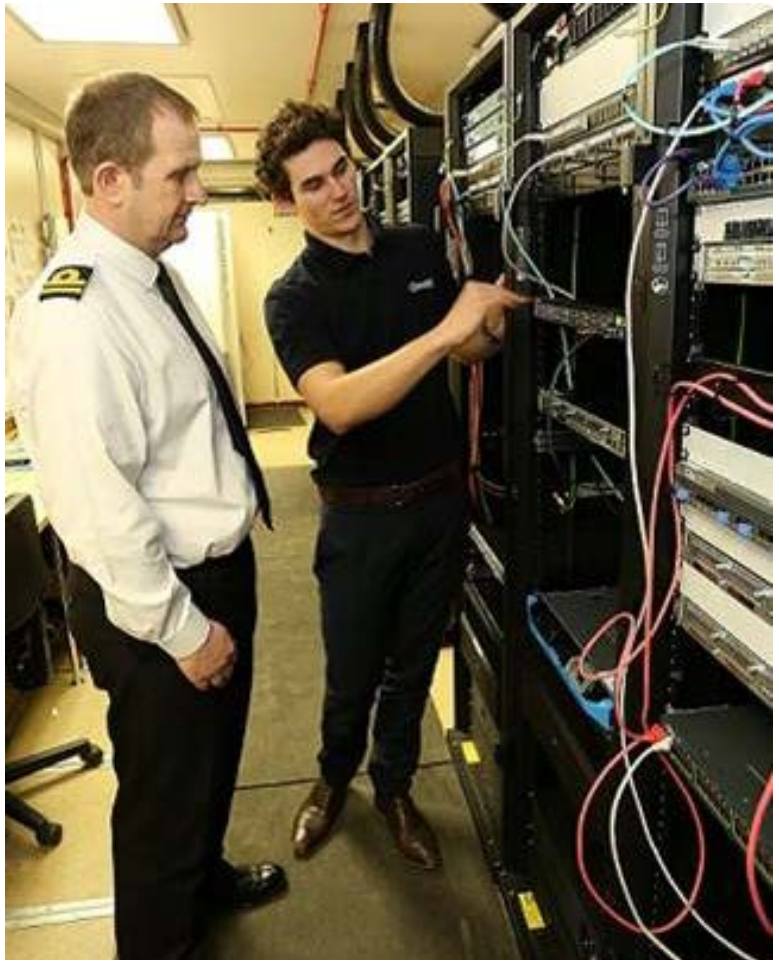
- We have shown that the Mission Planner AI engine can in principle be used to mount automatic cyber attacks.
- Although it was not possible to connect the optimizer directly to the attacked network, this was simulated by a script marker.
- Whilst suitable for a demonstration, this does limit the optimizer, because an important advantage of the approach adopted is that it is able to find novel solutions to problems that have never been solved before.
- It is not possible to demonstrate this using script marking assessment.



QINETIQ

7 Conclusions (2 of 3)

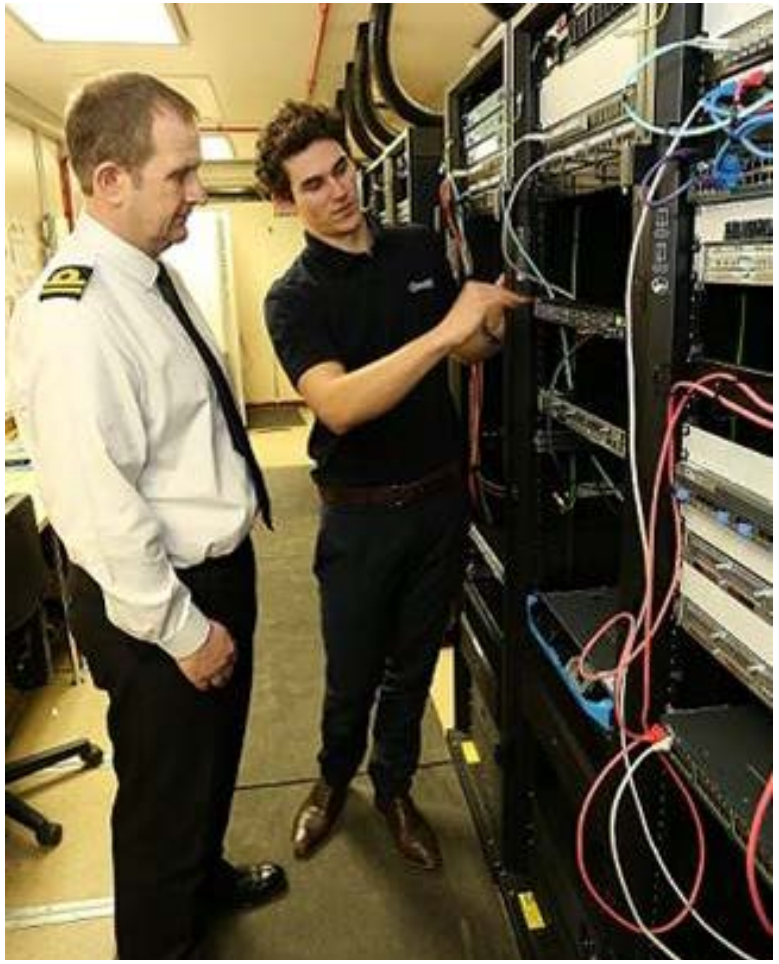
- The next step will be to replace the script marker by a Mininet, Containernet, libvirt based solution.
- We believe that this will allow rapid prototyping of target environments by the optimizer.
- An important part of this work will be to assess the effect on run times.
- A wider range of target networks, and types of cyber attack, should also be tested.



QINETIQ

7 Conclusions (3 of 3)

- The Mission Planner AI engine could also be extended so that it can defend against cyber attacks.
- Attack and defence AI engines could be combined to form an adversarial AI in which the attack and defence components in turn are pitted against each other.
- Every pass through this attack-defence cycle will further improve the strength and robustness of both components as each is forced to respond to a progressively more capable opponent.



QINETIQ


QINETIQ