

Artificial Intelligence Applied to Cyber Modelling

George Skrobanski

QinetiQ Training & Simulation Ltd
Cody Technology Park, Old Ively Road, Farnborough, Hants, GU14 0LX
UNITED KINGDOM

george.skrobanski@nsc.co.uk

Nick McLauchlan

CyberPrism Ltd
20-22 Wenlock Road, London N1 7GU
UNITED KINGDOM

nm@cyberprism.net

ABSTRACT

In the modern military domain, timely information is increasingly critical. However, attacks are becoming more sophisticated. Quickly identifying, selecting and carrying out cyber defence responses is essential.

Novel Artificial Intelligence (AI) Statistical Forward Planning techniques have been implemented for conducting cyber-attacks on a target environment. These techniques have not been used before in cyber modelling, but their success in other domains suggests their utility. The AI demonstrates that it is able to assemble complex attacks, showing capability in cyber domain problems.

Significant progress has been made in the full representation of the target environment in a Mininet/Containernet solution, giving confidence that such representation is possible. This would allow building representative target environments for rapid analysis.

The AI could be used to generate data at large scale for signatures characteristic of successful attacks, facilitating training of Machine Learning AI to detect such attacks, critical where real-world data are sparse.

An advantage of this approach is that the AI is not learning but forward-planning. Therefore, as new exploits are identified, the attack and defence actions available to the AI can be easily updated, without need for retraining. This agility is critical in the rapidly evolving Cyber domain.

1.0 INTRODUCTION

Computer networks in the military domain, and everywhere, are increasingly subject to cyber attacks. It is clearly important to be able to quickly detect vulnerabilities to such attacks and select suitable responses. There is therefore 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, ref [1], 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.

By contrast, this paper describes an automated AI cyber simulation using statistical forward planning rather than RL. Statistical forward planning (SFP) algorithms are a family of robust stochastic AI techniques that

use a statistical model (also called a forward model) to simulate possible future states. They operate without training and are for this reason much faster than learning-based methods. Example SFP algorithms include Monte Carlo Search, Monte Carlo Tree Search, and Rolling Horizon Evolution. For a recent reference, see Perez-Liebana *et al*, ref [2], which describes the application of SFP to a video computer game. Rapid algorithms, and games, are of course also of importance in the military context.

The AI software used in this project is an extension of the Mission Planner AI tool developed by QinetiQ Training and Simulation Limited. 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 anti-submarine 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. In particular, Mission Planner can be used to plan cyber attacks and defences. See ref [3] for a discussion of Mission Planner in its original land-based war game persona.

The AI engine within Mission Planner uses stochastic optimization algorithms (in this case simulated annealing) to select an optimal sequence of orders. Or, in the cyber context, 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.

The aim of the work was to determine whether Mission Planner can be used to generate cyber attacks on computer networks. It is important to keep in mind that the present paper describes only the first stage of this work.

2.0 APPROACH

2.1 Cyber Attack Structure

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

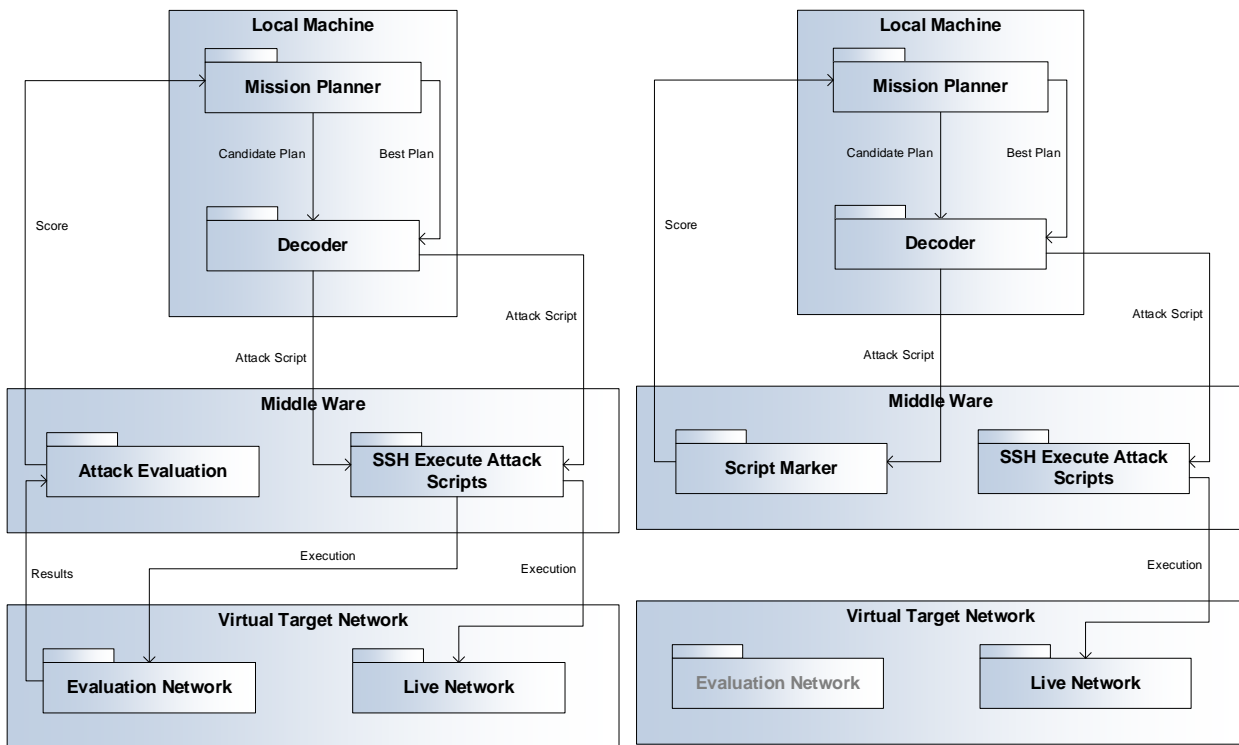


Figure 2-1: Cyber Attack Structure: as planned (left) and as implemented (right).

The intended structure is shown in the left hand diagram in Figure 2-1.

- 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.

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, so was non-functional, it was not possible to implement the target network using Mininet. 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 right hand diagram in Figure 2-1.

2.2 The Script Marker

As explained in ref [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.

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.

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 dividing 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.

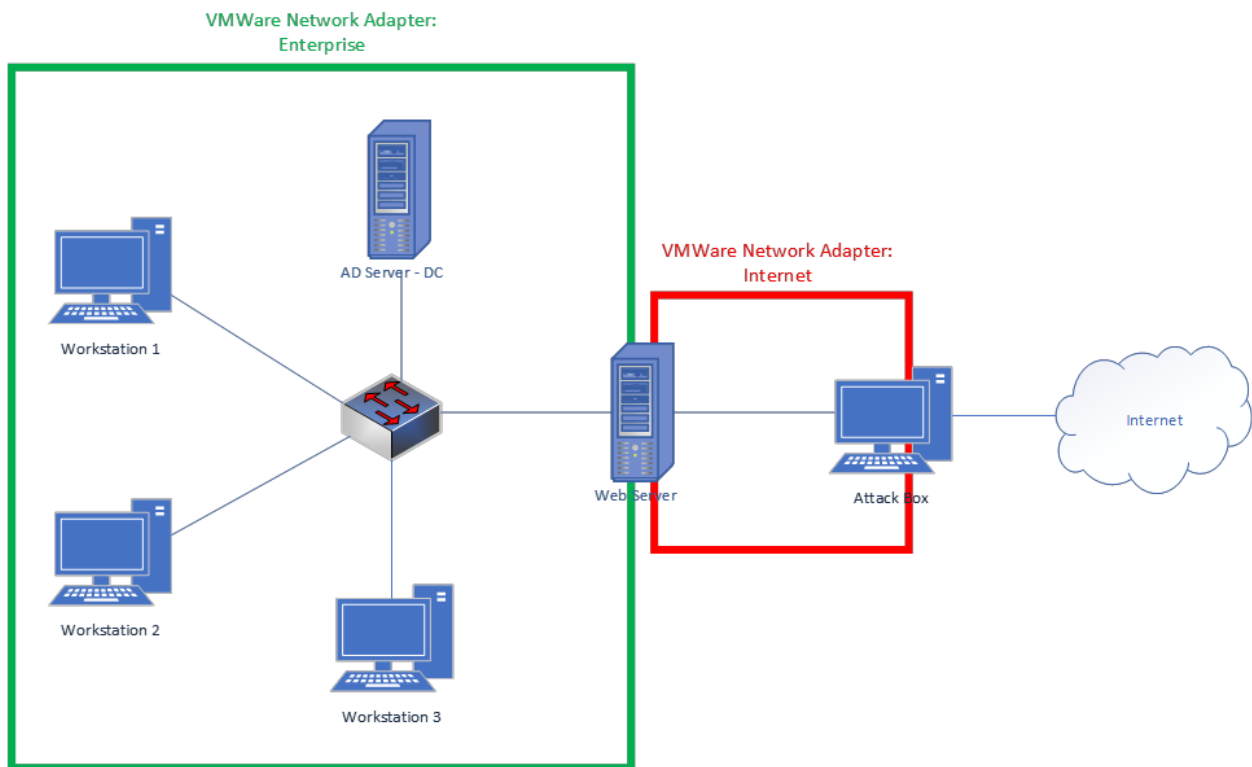


Figure 2-2: The live network: a complete enterprise architecture using VMWare.

2.3 The live network

Figure 2-2 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. See ref [4]. 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. See ref [5]. 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. See ref [6]. 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 of Fig 2-2 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.

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. See ref [7].

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.

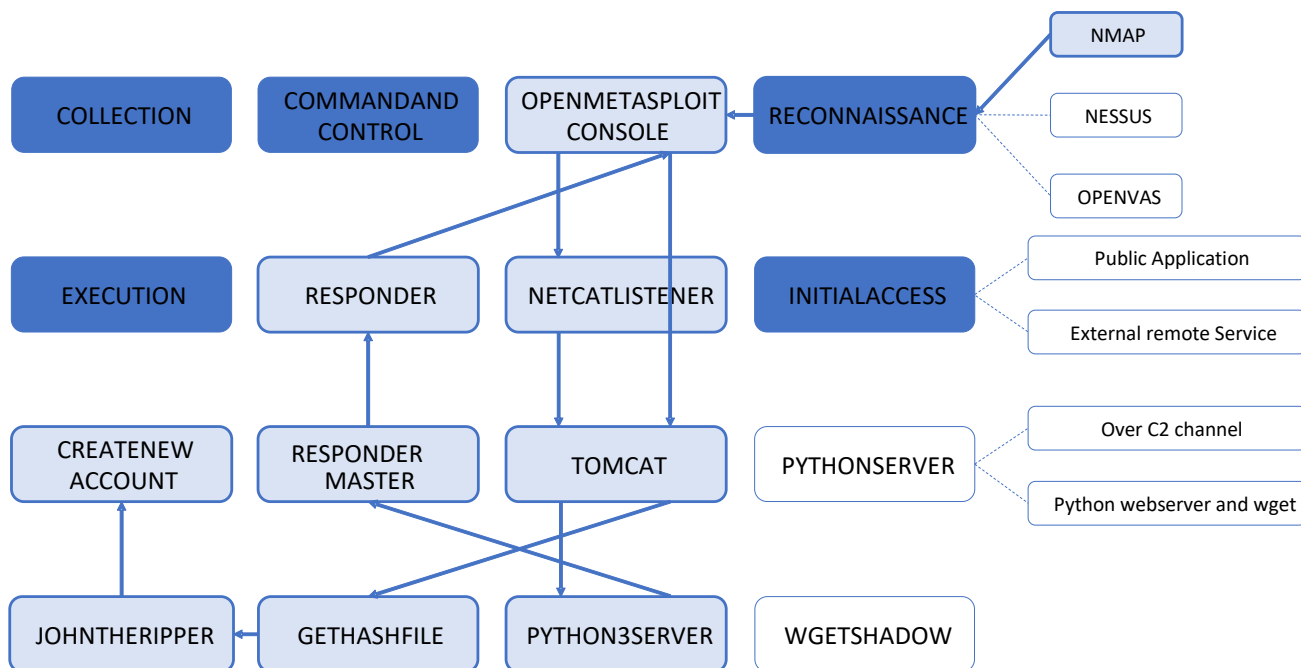


Figure 2-3: Attack script components required for domain take over.

2.4 AI attack syntax

Figure 2-3 shows the attack script components required for domain take over (tactics are shaded in dark blue, techniques are shaded in light blue), as well as some additional components that are not required for the current attack. Mission Planner can decide which scripts are required from a potentially limitless set of options. For example, the not 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. See ref [8]. These components are provided to Mission Planner which must then assemble them in the correct order to achieve the desired outcome.

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 Figure 2-3. The solid arrows indicate the correct sequence of script components required for the domain take-over attack.

3.0 AI PERFORMANCE

Table 3-1: AI progress reports for domain take-over script.

Generation	Best Score	Average Score	Largest	Average Size	Cumulative Time (sec)	Fraction replaced
1	66.667	30.578	51	13.705	1.59	0.70527
2	72.222	31.733	79	14.931	3.40	0.69357
3	74.779	30.1	178	21.131	5.79	0.70196
4	74.779	33.219	66	14.805	7.44	0.6821
5	74.779	34.181	54	14.533	9.15	0.65849
6	74.779	34.214	55	15.028	11.27	0.66099
7	74.9	37.34	40	12.779	13.19	0.62636
8	74.9	39.803	34	13.049	14.73	0.60488
9	80.455	41.154	29	12.591	16.61	0.58879
10	80.556	43.014	33	12.573	18.46	0.58035
11	80.556	45.291	31	12.333	20.13	0.55896
12	84.363	47.624	37	12.738	21.94	0.54465
13	84.363	50.407	22	12.229	23.51	0.52152
14	86.764	54.748	19	12.019	25.05	0.47969
15	93.994	56.303	18	12.079	26.46	0.45686
16	93.994	61.159	19	11.917	28.26	0.39016
17	93.994	63.515	17	11.925	30.14	0.32042
18	99.121	70.429	15	11.919	31.71	0.20595
19	99.674	75.678	15	11.899	33.56	0.098688
20	99.992	88.921	14	11.994	35.32	0.018495
21	99.999	86.567	14	11.973	37.17	0.024843

Table 3-2: Key to AI progress reports.

Generation	Each generation considered 15,200 attack scripts.
Best Score	The best similarity score in the current generation, expressed as a percentage.
Average Score	The average similarity score in the current generation, expressed as a percentage.
Largest	The number of components in the longest script in the current generation.
Average Size	The average number of components in each script in the current generation.
Cumulative Time (sec)	The elapsed time since the start of the optimization process.
Fraction replaced	The fraction of candidate scripts accepted by the simulated annealing algorithm in the current generation.

Table 3-1 lists Mission Planner’s AI progress reports for the domain take-over script, and Table 3-2 explains the contents of each column. The Table 3-1 shows that at generation 15 the optimizer achieved a best score of 93.994%. This script differed from the target script by only step, although it still resulted in a failed attack. At generation 21 a near perfect solution is obtained, resulting in a successful attack. The total elapsed time was 37 seconds.

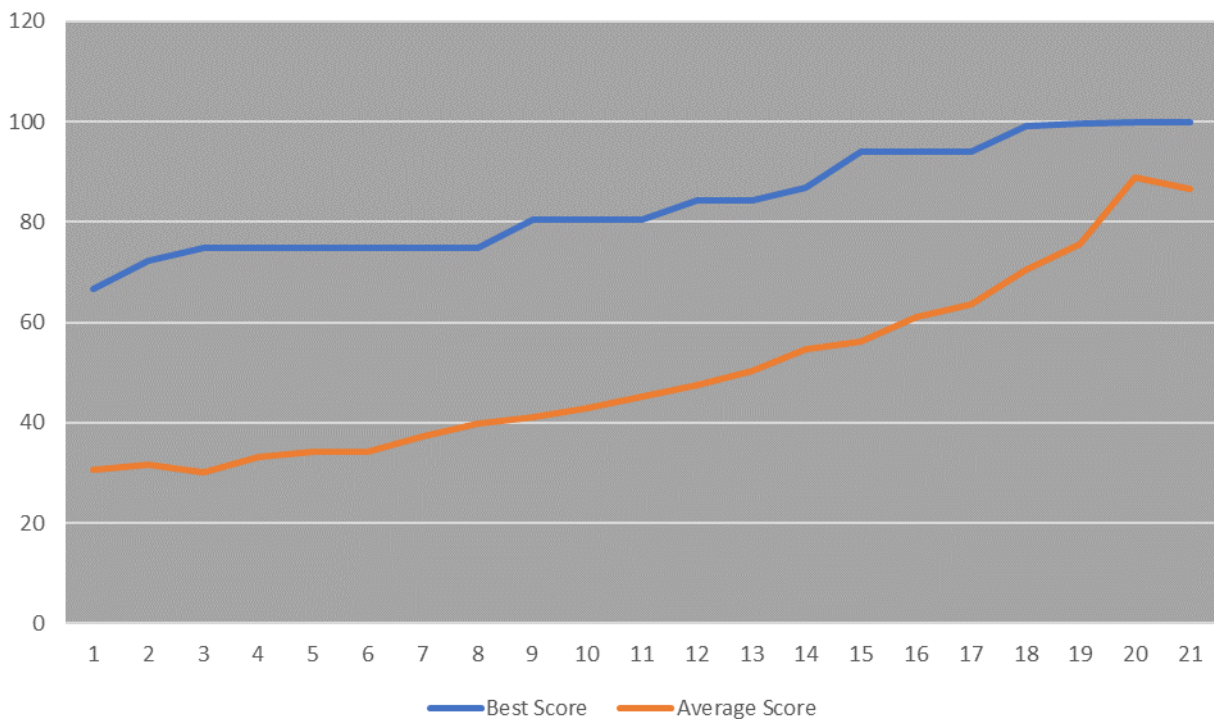


Figure 3-1: Improving percentage score during domain take over.

Figure 3-1 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.

4.0 CONCLUSIONS

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.

The next step will therefore 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.

The Mission Planner AI engine could also be extended so that it can defend against cyber attacks. Once this has been implemented, the 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.

REFERENCES

- [1] CyberBattleSim, Microsoft Research, <https://www.microsoft.com/en-us/research/project/cyberbattlesim/>
- [2] Analysis of Statistical Forward Planning Methods in Pommerman. Diego Perez-Liebana, Raluca D. Gaina, Olve Drageset, Ercument Ilhan, Martin Balla, Simon M. Lucas. Proceedings of the Fifteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-19)
- [3] Using artificial intelligence algorithms for high level tactical wargames and new approaches to wargame simulation. Lucek, S.G., Collander-Brown, S.J., 2017 JAOR, Vol. 9, No. 1
- [4] The Cyber Kill Chain <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>
- [5] NTLM Overview <https://docs.microsoft.com/en-us/windows-server/security/kerberos/ntlm-overview>
- [6] John the Ripper <https://www.openwall.com/john/>
- [7] MITRE ATT&CK® <https://attack.mitre.org/>
- [8] ZeroTier <https://www.zerotier.com/>