Automated Computer Network Exploitation with Bayesian Decision Networks *

Graeme M. Roberts, Gilbert L. Peterson
Air Force Institute of Technology
graeme.roberts.1@us.af.mil, gilbert.peterson@afit.edu

Abstract

Penetration Testing (pentesting) is the process, tactics and techniques to penetrate computer systems and networks to expose cybersecurity issues. It is currently a manual process requiring significant experience and time that are in limited supply. One way to reduce time is through automation. This paper presents the Automated Network Discovery and Exploitation System (ANDES) which demonstrates the feasibility to automate the pentesting process. Uniqueness to ANDES is the use and updating of Bayesian decision networks to represent the pentesting domain and subject matter expert knowledge and processes. Each iteration begins by modeling the current belief state for each system using a Bayesian decision networks. ANDES uses these networks to select and execute an expected best action. This process simulates the iterative thinking process of human attackers as they access and move through an enterprise network. Testing used a virtual network environment designed to mimic a small businesss internal network. ANDES successfully performed a series of information gathering and remote exploit actions, across multiple network hosts, to gain access to the objective target.

Introduction

Penetration testing (pentesting) has become an increasingly important part of an organization's security toolbox. Pentesting provides companies a way to determine their exposure to threat actors, in a proactive manner, that can ultimately allow the company to save time, money and reputation (RSI Security 2020b). A formal recognition of this importance is the adoption of frequent pentesting as a requirement in the Healthcare Insurance Portability and Accountability Act (HIPPA) as well as in many industry security standards such as the Payment Card Industry Data Security Standard (PCI DSS).(RSI Security 2020a) A major factor limiting broad adoption of pentesting by companies is the lack of available security professionals capable of conducting high-quality pentesting events. Combining automation with skilled professionals is a great way to increase capacity and alleviate talent shortages (Craig 2019).

*The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. The authors would like to thank Dr. Chad Heitzenrater AFRL/RIG for sponsoring this research.

Researchers have explored automated pentesting over the past two decades, however it has never transitioned towards mainstream adoption (McKinnel et al. 2019). One factor contributing to the lack of adoption is the complexity of the decision space and dynamic nature of pentesting scenarios. Previous automation attempts have relied upon modeling entire networks and completely solving an attack plan prior to execution (McKinnel et al. 2019). This approach is not representative of the iterative approach employed by human penetration testers, and is only capable of showing static results. Pentesters leverage tools and techniques to gain valuable information throughout their test, which informs future decision making. The development of the Automated Network Discovery and Exploitation System (ANDES) is an attempt to create an automated pentesting system that aligns with the human pentester process.

The Automated Network Discovery and Exploitation System (ANDES) is a proof of concept system that demonstrates how Bayesian decision networks can be employed to enable automated live-execution pentesting. The Bayesian decision networks capture the pentesting environment, as well as a human decision making process. Utilizing Bayesian decision networks allows the system to account for domain uncertainty and probabilistic actions.

Additionally, ANDES utilizes an iterative execution approach. Each execution cycle ANDES undertakes is an isolated event whose observed outcome is utilized to inform future decisions. In this way, ANDES is able to react to an uncertain environment and mimic the human decision making process of first gathering information, deciding what is the best course of action, and finally taking that action and observing the outcome.

Background

The vulnerability assessment and pentesting (VA-PT) processes (PTES 2014) can be resource intensive as it requires highly trained and experienced personnel (Craig 2019). These requirements led researchers to explore the idea of automating VA-PT actions, using computer based systems to either augment or replace trained security professionals. A major challenge facing automated VA-PT systems is their ability to capture the Subject Matter Expert (SME) knowledge of security professionals, as well as their ability to synthesize observed information into optimal decision policies.

The capability to adapt to the observed state of the domain proves particularly difficult and is a major focus addressed by this work.

Penetration Testing Automation

Previously explored solutions can be divided into the categories of: classical planning (constraint satisfaction), probabilistic planning, adversarial planning (min-max), live-execution, and adaptive planning (learning-models). McKinnel et al. (McKinnel et al. 2019) present an overview of existing research and advocate for research to focus on testing via live network interactions.

One attempt at incorporating planning under uncertainty concepts into a usable product was in (Obes, Sarraute, and Richarte 2010). The authors integrate the Metric-FFplanning system with the *Core Impact* pentesting tool. The accomplishments include the ability to convert the internal state of Core Impact into PDDL and convert the attack plan output into a Core Impact format. One drawback is that this system is provided full knowledge of the entire network and generates an attack plan designed to traverse the network assuming deterministic outcomes. Realizing a weakness of the system came from the inability to account for uncertainty, led the authors to pursue using POMDPs in place of the planner (Sarraute, Buffet, and Hoffmann 2012; 2011). Unfortunately the authors never integrated the POMDPs into a system that interacts with a live network.

In the commercial sector, the current version of Core Impact offers RPTs which automate portions of the VA-PT process (Core Security). An RPT is given a computer system as a target and it automatically conduct information gathering actions, develop an attack profile, conduct those attack steps and ultimately report the results. The system does not use gathered information or access to pursue additional network targets and simply provides a summary of the results as the output. The system does not attempt to identify a 'best' point of access to the system, but merely attempt all exploits meeting a threshold included with the rule-set (level of risk, type of exploit, etc.).

A difficulty of creating systems that operate in real-world environments is the creation of the system's domain representation. The starting state of the network, required for most of the systems, is externally produced and provided to the automated systems (McKinnel et al. 2019). (Ghazo et al. 2019) addressed this by leveraging commercial network vulnerability scanning products. However, these products rely on having administrative credentials and knowing the layout of the target network. The results of the commercial scanning products were then imported into a model building system to create a pentest plan. This solution is not applicable to unknown environments or non-cooperative VA-PT scenarios.

Bayesian Decision Networks

Bayesian decision networks are probabilistic directed acyclic graph representations of elements in a domain and decision actions with their conditional dependencies (Matzkevich and Abramson 1995). Bayesian decision networks contain three types of variables: chance variables $\mathbf{V}_C$, decision variables $\mathbf{V}_D$, and utility variables $\mathbf{V}_U$. Chance variables are those that represent domain states the decision maker has no direct control over. Decision variables are those which represent actions directly controlled by the decision maker. Lastly utility variables represent the decision maker's preferences. Since variables are represented as nodes within the graphical representation, the terms variable and node are used interchangeably (Lacave, Luque, and Dez 2007). Useful definitions include a *finding* (f), which corresponds to a value a chance variable can take on ($V_C = f$).

The graph structure of a Bayesian decision networks consists of nodes connected via directional arcs. Every node must be connected to at least one other node and no cycles may exist. An incoming arc's meaning is determined by the type of connected node:

- Arcs into a decision node represent information known at the time of the decision
- Arcs into a chance node represent probabilistic dependence
- Arcs into a utility node represent functional dependence, i.e., which node values are used to calculate utility

Bayesian decision networks are often employed to help produce consistent and mathematically sound decision making. For example, Elvira(Lacave, Luque, and Dez 2007) leverages a Bayesian decision network to help teach medical decision makers probabilistic and repeatable decision making skills (Lacave, Luque, and Dez 2007).

ANDES

ANDES consists of three major components. The first component is the Control Component which guides execution and coordinates information flow between the other two components. Second is the Decision Component which receives the Control Component's current belief state of the target domain and decides the next action for execution. The final component, the Execution Component interacts with an enterprise network to perform actions and sensing.

Control Component

The Control Component maintains and updates the system's internal belief state and domain knowledge. It uses these to guide system execution and coordinate information flow.

ANDES represents computer hosts and their connections as Host Objects. Host Objects are a custom data structure that capture relevant host details that include: current control state (control, proxy, scanned, active session) and host state (operating system, hostname, IP address, services, domain, neighbors, and scan results). There is one Host Object per host encountered in a network.

The execution cycle consists of seven steps shown in Figure 1.

Step 1 begins when the Decision Component receives the system state from the Control Component. The Decision Component creates a Bayesian decision network for each Host Object and calculates the MEU as well as the associated decision(s). Step 2 utilizes selection logic to determine which target and action combination is expected to yield

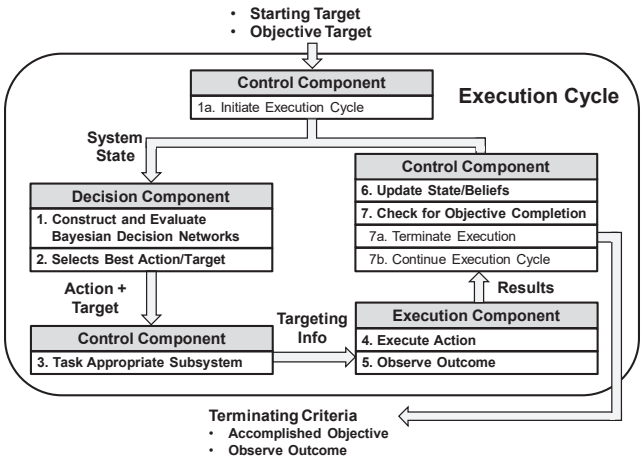

Figure 1: The Control Logic flow of execution.

the greatest utility. These results are returned to the Control Component. In Step 3, the Control Component determines what targeting information is required and passes it to the appropriate Execution Component subsystem.

The Execution Component consists of two subsystems. The Scanning Subsystem conducts network scans and the Exploitation Subsystem conducts local information gathering actions on compromised targets (post initial compromise). Once the action is prepared the Execution Component launches the action and prepares to observe the results. Step 5 consists of observing and processing the results to determine the actual action outcome and passing them back to the Control Component.

Results Analysis updates any applicable Host Objects or the overall system state. This step also performs internal status checks and updates. An example is 'pinging' all target hosts. If the connection responds appropriately no action is required, however if the hosts do not respond appropriately ANDES assumes the connection has somehow been interrupted and updates the target's Host Object accordingly.

Step 7 uses the updated system state to determine whether ANDES has reached the user defined objective. If the objective has not been reached, Step 7b restarts the cycle. If the objective has been reached, Step 7a will terminate execution, alert the user to objective completion and conduct required shutdown actions.

Execution Cycle Advantages The iterative process enables ANDES to take advantage of information gained during execution, as well as to account for unexpected results, much like an expert. An added benefit of this method is the decision space (and corresponding solution space) is kept small allowing for efficient solving. The size of any specific Bayesian decision network never grows, only the number of networks generated grows. This means the decision making time grows linearly as the target network grows.

Decision Component

The Decision Component performs reasoning and utilizes Bayesian decision networks to model each individual decision. This is in contrast to previous solutions which represent the entire decision space and solve for an entire attack chain, starting from target information and terminating at objective completion (McKinnel et al. 2019). ANDES selects a single action at each decision point, which it then executes, observes the outcome, updates the belief state and repeats. This method allows for easy handling of non-deterministic actions as well as accounting for domain uncertainty and unknowns.

The Decision Component (Figure 2) begins when the Control Component provides the Decision Component with all currently existing Host Objects. The Decision Components generates a unique Bayesian Decision Network for each Host Object. Next, each network is solved and the maximum expected utility (MEU) found. The Decision Component selects the action/target pair associated with the highest MEU and return the best next action for ANDES to take.

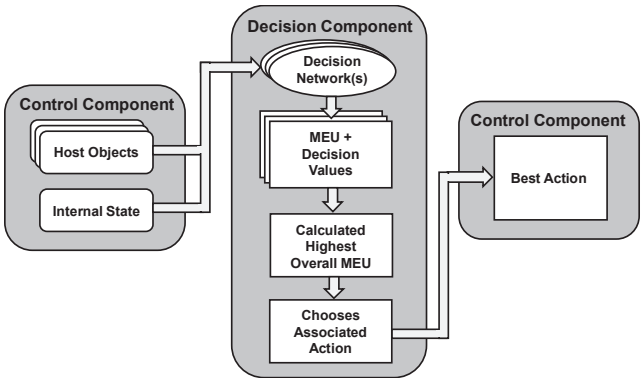

Figure 2: The Decision Component process.

Decision Networks

ANDES represents each host as its own Bayesian decision network (shown in Figure 3), populating each network with observed evidence from ANDES's current belief state of that target. The network was developed by a VA-PT expert and refined from feedback from two additional expert practitioners.

Chance nodes ($\mathbf{V}_C$)(ovals): Capture the probability of a variable and are present in two distinct types:

- Observed Chance Nodes: Directly capture observable attributes of the host and are populated during network initiation.
- Contingent Chance Nodes: Capture SME domain knowledge regarding expected outcomes, given observed evidence and potential decisions.

The 'compatible exploit' chance node is an example of a contingent chance node as it combines the observed evidence for the operating system and vulnerabilities to calculate the probability of whether ANDES contains a compatible exploit given observed evidence. In this way it introduces outside knowledge regarding exploit and operating system compatibility, as well as potential chances of success given various operating systems and vulnerability combinations.

The network contains 13 chance variables:

$$\mathbf{V}_C = \{OS, Vul1, Vul2, Vul3, Scan, Neighbor, Access, TN, SR, AR, CE1, CE2, CE3\}.$$

Of these 13 variables, 8 are observable during execution:

- $OS = \{Win7, Win2k19, Linux, Unknown\}$: Observed Operating System
- $Vul1/2/3 = \{T, F\}$: Does the corresponding vulnerability potentially exist?
- $Scan = \{None, Unfiltered, Filtered\}$: Has the target been scanned and were any results likely to have been filtered?
- $Neighbor = \{T, F\}$: Does the target have a neighbor within the network ANDES has access to?
- $Access = \{T, F\}$: Does ANDES have access to the host via a Metasploit session?
- $TN = \{T, F\}$: Is the target within ANDES's target's subnet?

The remaining five chance variables, represent contingent chance variables.

- $SR = \{None, Unfiltered, Filtered\}$: If the target is scanned, what is the likelihood additional information will be gained?
- $AR = \{T, F\}$: Given the chosen actions and current state, what is the likelihood ANDES will have access to the host post execution?
- $CE1/2/3 = \{T, F\}$: Given observed Operating System and potential vulnerability, what is the likelihood ANDES has a compatible exploit?

Decision nodes ($\mathbf{V}_D$)(rectangles): Represent potential decisions ANDES could make. Networks contain two decision nodes corresponding to whether ANDES should acquire information (scan) or acquire new access (exploit). The two decision variables are:

$$\mathbf{V}_D = \{AD, ED\}$$

. Within ANDES these represent:

- $AD = \{Exploit, Scan\}$: Should ANDES conduct an information gathering action or attempt to exploit the target?
- $ED = \{Exploit1, Exploit2, Exploit3\}$: If ANDES conducts an exploit action, which potential exploit should be chosen?

Utility nodes ($\mathbf{V}_U$)(diamonds): Evaluate the expected utility of a given network configuration. ANDES evaluates the expected utility for each combination of potential decisions for each host network. This utility function should be set by operators to account for preferences and desired behavior. For instance the utility node's values could be adjusted to value information more highly in an attempt to minimize taking risky uninformed actions. ANDES has a single utility variable:

- $AU = \psi_{AU}(SR, AR, Access, TN)$: What is the expected utility of the network's resulting state? Considers the current combination of known states, chosen actions, and calculated probabilities.

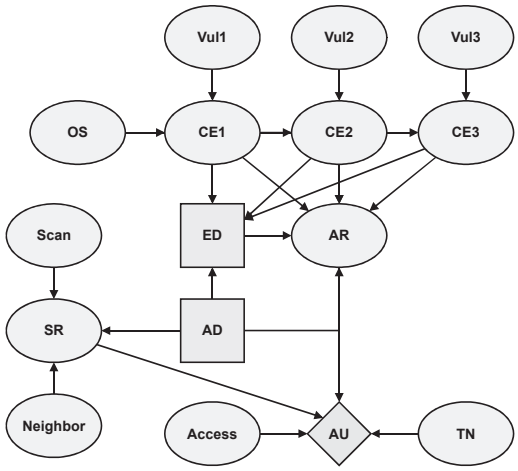

Figure 3: ANDES host Bayesian decision network.

Execution Component

The Execution Component consists of two subsystems. The Scanning Subsystem performs network scanning actions and the Exploitation Subsystem executes remote exploitation and local information gathering actions.

Scanning Subsystem The Scanning Subsystem scans remote hosts based on targeting information from the Control Component. Scanning employs the libnmap library(Lyon) to initiate an Nmap scan, storing and parsing the results as they return. The parsed scan results include a host's hostname, operating system, open ports and associated services. The information is embedded into a Host Object.

The Scanning Subsystem also supports scanning remote targets that have been exploited via a remote proxy. This enables ANDES to leverage access inside of a target network for improved information gathering and to bypass external security measures.

Exploitation Subsystem To enable exploitation actions ANDES interfaces with a Metasploit-4.19.0 RPC server (Rapid7). Metasploit is an industry standard, publicly available exploitation framework. Metasploit contains both an exploitation and post-exploitation framework enabling remote exploitation attacks, payload delivery, and interactions with compromised hosts.

The Metasploit RPC client creates a session object containing a handle to the active session residing on the RPC server. A reference to this session object is stored within the associated target's ANDES host object, tying the target and session together. This enables the Exploit Subsystem to interact with active sessions in the future. Besides the host survey, the connection provides the ability to establish a proxy pivot on the compromised host. This enables future scanning and future remote exploits to avoid external security measures.

A critical implemented component is an exploit execution feedback module. This module converts the real-world result of Metasploit execution into the host's belief state.

Testing and Results

Testing ANDES evaluates the viability behind the principles and concepts employed. This qualitative approach is in line with the avenues for potential research directions within (McKinnel et al. 2019) who advocates that systems should both move towards live-execution testing and a more qualitative approach of assessing the systems performance regarding the benefits of the system.

Testing demonstrates the ability of ANDES to perform a full penetration test on a simplified real-world problem a penetration tester might face. ANDES was tasked to traverse the target network in search of the **FileServer** host, enabling future data exfiltration from the target.

In order to accomplish this goal, an SME would conduct scans of the target network, identify potential vulnerabilities, exploit those vulnerabilities, bypass security measures, and create pivots to gain further access into the enterprise until locating the target. ANDES is designed to capture an SME's knowledge and utilize it in unknown environments to reproduce their decision making process and skills. As such, evaluation looks at if ANDES makes the same choices as the SME whose knowledge was encoded into the system.

Test Environment

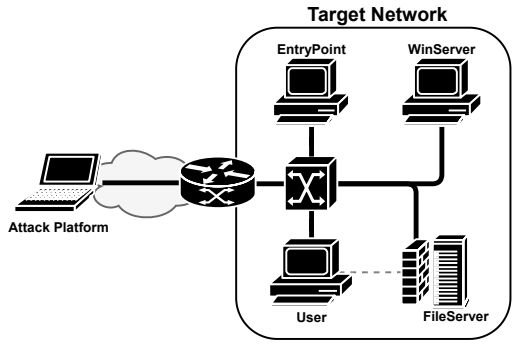

Figure 4: The evaluation network. The dotted line represents a Firewall rule that restricts access to the **FileServer** host from the **User** host.

The evaluation network is a virtual environment designed similar to an environment a pentester might encounter at a small business (Figure 4). The network consists of four possible targets, each with a unique configuration. The network contains:

- Two user machines running Windows 7 corresponding to employee workstations.
- Machine running Windows Server 2019 and acting as the Windows Domain Controller and internal DNS Server.
- Ubuntu host serving as a file server, utilized by the business to store company data.

Every host has a vulnerability to at least one of the exploits available to ANDES with the exception of the **WinServer**. The **WinServer** host is running a vulnerable service, however the advanced protections included in the Windows Server 2019 Operating System prevent the service from being exploited. This serves to show how ANDES reacts to non-vulnerable hosts.

Attack Chain Results

Table 1 shows the entire series of actions ANDES performed. During initiation of the test event ANDES was provided with only two pieces of information, the IP Address of the **EntryPoint** host and the IP Address of the **FileServer** as the objective target.

ANDES began execution by conducting a remote network scan of the initial target (**1**). ANDES detected the **EntryPoint** host was running the vulnerable *Dup Scout Enterprise* service, as well as being potentially vulnerable to the *EternalBlue* exploit. The SME knowledge captured within ANDES's Bayesian decision network led ANDES to chose to attempt to remotely exploit the *Dup Scout Enterprise* service as it had a higher chance of success (**2**). The action was successful and ANDES gained access to the **EntryPoint** host. Upon gaining access, ANDES conducted a host survey to discover additional targets (**2.1**).

The host survey resulted in ANDES discovering that **EntryPoint** had connectivity to the **WinServer** host and **User** hosts. The next action selected was to perform a remote network scan of the **WinServer** host (**3**). Given the fact ANDES did not have any additional information besides the hosts' IP Addresses at the time, and neither host was an objective, selecting to scan either of the two new targets represents a reasonable decision. From the scan ANDES determined the **WinServer** host was also running the vulnerable *Dup Scout Enterprise* service. With this information, and no additional information about the **User** host, ANDES attempts to exploit the **WinServer** (**4**). As expected given the **WinServer** host's operating system protections, the exploit was unsuccessful, which resulted in ANDES detecting action failure.

Following this unsuccessful exploit attempt, ANDES decided to scan the **User** host (**5**). This scan revealed the **User** host may be susceptible to the *EternalBlue* exploit. ANDES then decided to attempt to the *EternalBlue* exploit (**6**), which was successful and ANDES gained access to the **User** host, after which it conducted a host survey (**6.1**).

This host survey revealed that the **User** host has connectivity to the **FileServer** host, ANDES's objective. Additionally, the access afforded by *EternalBlue* is sufficient to allow ANDES to establish a Pivot on the host. ANDES creates a pivot which forwards traffic to the **User** host's neighbors (**7**), in this case the **FileServer** host. Having located the objective host ANDES performs a remote network scan of the **FileServer** target (**8**) through the **User** host pivot. This scan revealed the fact that the SSH service is reachable on the **FileServer** from the **User** host. ANDES then chose to launch an *SSH Brute Force* attack against the **FileServer** through the **User** host pivot. The first two attempts at this exploit fail (**9+10**). ANDES continues to attempt this exploit until it is eventually successful (**11**). ANDES has now reached the desired objective host and without further objectives terminates all active connections and shuts down (**12**).

An item to note is ANDES's willingness to repeatedly attempts at exploiting the **FileServer** host. This is in contrast to the previously failed exploit against the **WinServer**

Table 1: System Performance Test Results.

#	Action	Target	Results	Comments & Objective Demonstrated
1	Scan	EntryPoint	Success	Initial Target
2	Exploit1	EntryPoint	Success	Dup Scout BOF - Obj 1a
2.1	HostSurvey	EntryPoint	Success	Added WinServer, User to targets- Obj 2
3	Scan	WinServer	Success	
4	Exploit1	WinServer	Failure	Dup Scout BOF - Obj 3
5	Scan	User	Success	
6	Exploit2	User	Success	EternalBlue - Obj 1b
6.1	HostSurvey	User	Success	Added FileServer to targets- Obj 2 Found Target Objective
7	Pivot	User	Success	Added pivot to internal network through 'User'
8	Scan	FileServer	Success	Scan conducted through pivot - Obj 4
9	Exploit3	FileServer	Failure	Attack conducted through pivot - Obj 4, Obj 3
10	Exploit3	FileServer	Failure	Attack conducted through pivot - Obj 4, Obj 3
11	Exploit3	FileServer	Success	SSH Brute Force - Obj 1c. Target Objective Reached
12	Shutdown	N/A	Success	Performed clean-up actions

host, after which ANDES proceeded to pursue a different action. This aligns with SME decision weighting to favor taking actions against the objective host. It should be noted that a brute force logon attempt is not normally an exploit that is commonly repeated, but in virtual environments, coupled with the behavior of the *Metasploit SSH Login Scanner* through a proxy, this attack routinely takes several attempts.

Conclusion & Future Work

ANDES represents a shift in the design philosophy towards automated pentesting systems. Representing each host within a target network as Bayesian decision networks addresses previous issues with computational complexity as the target network size grows. This choice, along with chaining together isolated decision points, allows ANDES to mimic the decision making process employed by human SMEs and handle unforseen situations. This was demonstrated with ANDES successfully performing the pentesting steps: intelligence gathering, threat modeling, vulnerability analysis, exploitation, and post exploitation.

ANDES's currently contains the minimum functionality required to conduct automated pentesting events. Future research should include building a robust and complex Bayesian decision network that captures additional SME and domain knowledge. Additionally, integrating automated exploitation generation capabilities as demonstrated in DARPA's Cyber Grand Challenge (Song and Alves-Foss 2015) would lead to a more capable system.

References

Core Security. Core Impact's Rapid Penetration Tests. https://www.coresecurity.com/products/core-impact/rapid-pen-tests.

Craig, R. 2019. Closing the cybersecurity skills gap. https://www.forbes.com/sites/ryancraig/2019/11/26/closing-the-cybersecurity-skills-gap/?sh=74e5ad2474a8.

Ghazo, A. T. A.; Ibrahim, M.; Ren, H.; and Kumar, R. 2019. A2g2v: Automatic attack graph generation and visualization and its applications to computer and scada networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 1–11.

Lacave, C.; Luque, M.; and Dez, F. J. 2007. Explanation of Bayesian networks and influence diagrams in Elvira. *IEEE Transactions on Systems, Man, and Cybernetics*.

Lyon, G. Nmap. https://nmap.org/.

Matzkevich, I., and Abramson, B. 1995. Decision analytic networks in artificial intelligence.

McKinnel, D. R.; Dargahi, T.; Dehghantanha, A.; and Choo, K. K. R. 2019. A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. *Computers and Electrical Engineering* 75.

Obes, J. L.; Sarraute, C.; and Richarte, G. 2010. Attack planning in the real world. *SecArt'2010 at AAAI 2010*.

PTES. 2014. Ptes technical guidelines. Technical report, Penetration Testing Execution Standard.

Rapid7. Metasploit. https://www.metasploit.com/.

RSI Security. 2020a. What are the different types of pen testing? Technical report, RSI Security.

RSI Security. 2020b. What is the penetration testing execution standard? Technical report, RSI Security.

Sarraute, C.; Buffet, O.; and Hoffmann, J. 2011. Penetration testing == pomdp solving? *Proceedings of the 3rd Workshop on Intelligent Security (SecArt'11)*.

Sarraute, C.; Buffet, O.; and Hoffmann, J. 2012. POMDPs make better hackers: Accounting for uncertainty in penetration testing. *Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*.

Song, J., and Alves-Foss, J. 2015. The DARPA cyber grand challenge: A competitor's perspective. *IEEE Security & Privacy* 13(6):72–76.