Cyber range automation overview with a case study of CRATE

Tommy Gustafsson
Swedish Defence Research Agency

Jonas Almroth
Swedish Defence Research Agency

Cyber range automation overview with a case study of CRATE

Tommy Gustafsson[0000−0003−4672−0962] and
Jonas Almroth[0000−0002−1077−9981]

Swedish Defence Research Institute (FOI), Linköping, Sweden
https://www.foi.se
tommy.gustafsson@foi.se jonas.almroth@foi.se

Abstract. Cyber security research is quintessential to secure computerized systems against cyber threats. Likewise, cyber security training and exercises are instrumental in ensuring that the professionals protecting the systems have the right set of skills to do the job. Cyber ranges provide platforms for testing, experimentation and training, but developing and executing experiments and training sessions are labour intensive and require highly skilled personnel. Several cyber range operators are developing automated tools to speed up the creation of emulated environments and scenarios as well as to increase the number and quality of the executed events. In this paper we investigate automated tools used in cyber ranges and research initiatives designated to augment cyber range automation. We also investigate the automation features in CRATE (Cyber Range And Training Environment) operated by the Swedish Defence Research Agency (FOI).

Keywords: Cyber range · cyber range automation · automated tools · Cyber Range And Training Environment (CRATE).

1 Introduction

A cyber range is a specialized facility dedicated to cyber security where research experiments and training sessions can be executed in a controlled fashion. The basic concept of the cyber range has been used since the beginning of the millennium with early examples being the Emulab [49], DETERLab [39] and U.S. National Cyber Range [15].

In order to better counter threats against computerized systems, there is currently a need for an increased number of experiments and training sessions [1, 13, 35]. There is a need to shorten the time taken [13, 15], and decrease the resources needed [11, 13], to setup and execute cyber range events. There is also a need of larger and more complex environments [15, 31] and to increase the fidelity of the experiments and the training sessions [11]. Furthermore, there is a need to validate the emulated environments prior to executing events [13, 22].

To address these challenges, several cyber range operators are developing automated tools [11, 13, 35, 45, 51]. In this paper, we investigate the current status and research trends in automated cyber range tools. We also describe the architecture and tools of the cyber range CRATE, operated by FOI, as an example of a cyber range where automation has been integrated into the design.

The remainder of this paper is organized as follows. In Section 2, related work is presented, followed by a presentation of the cyber range CRATE in Section 3. In section 4, we describe the automated tools integrated into CRATE. Section 5 describes how these tools have been utilized to perform research and training. Section 6 contains a compilation of the automated tools identified in eleven cyber ranges. The paper is concluded with a discussion of the findings in Section 7 and conclusions in Section 8.

2 Related work

In 2013, Davis and Magrath presented 28 cyber ranges and network testbeds [14]. Eight of these were described to include some form of automated features. In 2019, Yamin, Katt and Gkioulos presented a literature review where 100 papers are analyzed [51]. Based on the analysis, the authors of the surveys identify a research trend towards automated cyber ranges starting in 2014. This automation trend is also identified in a survey presented by Karlzén, where a literature review covering 74 cyber ranges is described [25]. Interestingly, the latter actually utilized an automated tool to perform the survey.

In total, fourteen cyber ranges containing different automated tools were identified by the surveys. AIT Cyber Range in Austria incorporate automation to deploy virtual machines during *capture the flag* (CTF) events [16]. The same cyber range also incorporates a tool called *GameMaker* used as a scenario engine to automatically execute injects during cyber range events [27]. Melón, Väisänen, and Pihelgas describe the tool suite *EVE and ADAM*, used to provide situational awareness during exercises hosted in the cyber range used by Nato Cooperative Cyber Defence Centre of Excellence (CCDCOE) [30]. The cyber range used by CCDCOE also incorporate an automated availability scoring system used during exercises such as Locked Shields [34]. Kim, Mæng, and Jang describe multiple automated tools needed in the cyber range used to host a complex exercise called Cyber Conflict Exercise [24]. The described tools automate activities such as system deployment and configuration, flag updates, attack execution and various types of scoring. An automated system which utilize virtualization features to restore the emulated environment after the event is also mentioned. Pham et al. describe an automated tool called *CyRIS (Cyber Range Instantiation System)* used in the cyber range CyTRONE [33]. CyRIS deploys and configures systems and services in the cyber range. In [43], an automated tool capable of executing attacks in CyTRONE is described.

In the cyber range DETERLab, a tool called *MAGI (Montage AGent Infrastructure)* is used to automatically run tests [44]. Davis and Magrath also mention that DETERLab is able to automatically deploy environments based on abstract test definitions [14]. Hibler et al. describe the automated tools used in the cyber range Emulab to allocate hardware, configure networking and execute events

[19], and in [8], the tool *Linktest*, used to validate emulated environments in Emulab, is described. Vykopal et al. describe a tool called *PM Portal* used to automate the setup and control of cyber exercises in the cyber range KYPO in [48]. They also mention automatic scoring of cyber exercises and that attacks can be automatically executed. In the future, a capability to automatically prepare and execute cyber experiments will be developed [48]. Braje describes a tool called *ALIVE (Automatic Live Instantiation of a Virtual Environment)* which is implemented into the cyber range LARIAT [11]. The tool uses configuration files to automatically build and configure virtual machines to create emulated environments. ALIVE is also able to configure many standard network services, including directory services, email servers, websites and file shares.

Urias et al. address the question of how cyber ranges can meet the increasing demand for cyber training and testing, with the U.S. National Cyber Range (NCR) as a use case. One of the solutions proposed is to utilize automated range provisioning and configuration tools to set up the emulated environments [45]. Automation of the NCR is further investigated in [35], where an overview of the tool suite called *FACTR (Flexible, Automated Cyber Technology Range)* is provided. FACTR automates core testing processes and procedures including testbed creation, verification and validation, monitoring, data collection, load and user behavior modeling, testbed reconfiguration, reconstitution, and execution [35]. Another cyber range that is described as partly automated by Davis and Magraph is VSCTC (Virtual Cyber Security Testing Capability). Shu et al. describe the automation features incorporated in VCSTC as capable to deploy emulated environments and to run experiments [38]. Davis and Magrath also describe the cyber ranges SIMTEX, CAAJED and ATC CYDEST as partly automated [14]. However, no further details have been found about the automation features in these cyber ranges, why these will not be further discussed in this paper.

The surveys presented by Yamin, Katt, and Gkioulos [51] and by Karlzén [25] also include several research initiatives where automated concepts and tools, not affiliated to any named cyber range, are presented. Russo, Costa and Armando introduce a scenario definition language used for scenario design and validation in [36]. In [13], the work is carried on with a description of a framework used for automating the definition and deployment of complex cyber range scenarios, based on a scenario definition language called VSDL (Virtual Scenario Description Language). VSDL will be integrated into the future Italian national cyber range [13]. A framework related to VSDL is also presented in [37]. In [12], Burke and van Heerden describe how automated attack capabilities can be developed for use in a cyber challenge environment. Abbott et al. describes how performance assessment can be achieved using automated parsing of log files generated during cyber training exercises [1]. Finally, Yasuda et al. present a tool called Alfons that automates the setup of an emulated environment using definition files and virtualization [52].

3 CRATE - Cyber Range And Training Environment

In this section, we describe the cyber range CRATE operated by the Swedish Defence Research Agency (FOI). The description is based on technical reports released in Swedish by FOI.

3.1 History

The development of CRATE started in 2008 and the cyber range has since been used in numerous research experiments such as [41], multiple training sessions [6], and exercises such as the Baltic Cyber Shield [20] and SAFE Cyber [47].

From the start, CRATE was developed to be a highly flexible and cost-effective cyber range, able to emulate large and complex environments [4]. To achieve this, automation has always been a priority, as exemplified by [17]. In 2016, development of a second generation of CRATE was initiated, where the lessons learnt from operating the cyber range are incorporated [4]. The second generation is scheduled to become fully operational in 2021.

3.2 Architecture

CRATE is a cyber range of the emulation type as categorised by Davis and Magrath [14], using both virtual machines and hardware devices. The research experiments and training sessions are conducted by running scenarios in emulated environments. Both the scenarios and the emulated environments are created and controlled with a set of cyber range tools developed by FOI. CRATE runs on a dedicated hardware platform that is hosted locally, a design choice made to ensure the flexibility and independence of the cyber range as well as the capability to handle sensitive data during research and training [4].

Figure 1 shows a high level architecture of CRATE with the *virtualization servers* that house the emulated environments in the center. The *control plane*, to the left, is utilized for cyber range management and the *event plane*, to the right, for the systems where the experiments and training sessions are executed. The planes represent two security zones and are isolated from each other, which is essential to ensure that the control plane is not affected by the events executed in the event plane.

The virtualization servers house the virtual machines used in the *emulated environments* (subsection 3.3). There are currently more than 500 virtualization servers operational in CRATE. The virtualization servers run a tiny, customized Linux-based operating system called CrateOS [5]. To facilitate cyber range maintenance and to ensure server integrity, CrateOS runs in a read-only environment and overlay file systems are used to store the virtual machines and configurations. This enables the operating system of the servers to be replaced without affecting the hosted virtual machines or their configuration, allowing CrateOS to be updated as new software versions and security updates become available. The process to update the servers has been automated using scripts, allowing the cyber range administrators to run the desired version of CrateOS on each

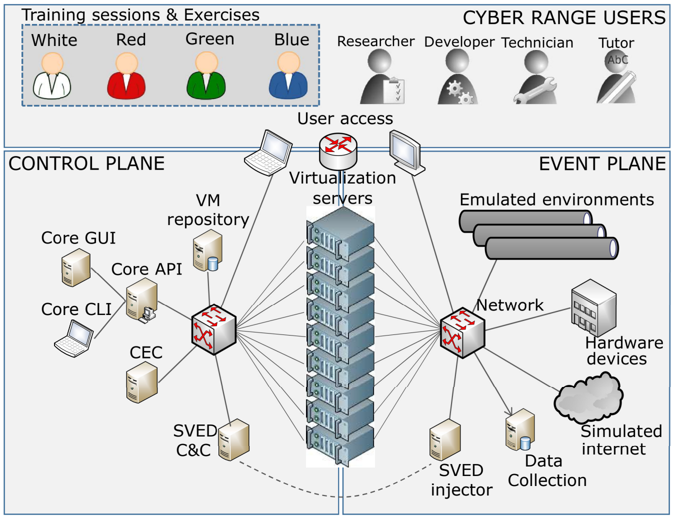

Fig. 1. High level architecture of CRATE showing the principal elements of the cyber range.

server. This capability further increases the cyber range stability and security [5].

Integrated in CrateOS is also a system service called NodeAgent. NodeAgent handles communication between the Core API and the virtual machines and automates the deployment and configuration of the emulated environments, as described in subsections 4.1 and 4.2.

There is a separate network infrastructure for each plane in CRATE. The LAN in the event plane is described by Almroth in [3] and utilizes software defined networking (SDN) to facilitate automated configuration of the emulated networks. VXLANs, virtual network segments, are used to support a high number of emulated networks. The VXLANs are dynamically assigned to the virtual machines' network cards and the routing protocols OSPF, RIP and BGP are used to share the routing information of each emulated environment within the event plane. Automatic management of the network configuration decreases the work load and skill required to create emulated environments in CRATE. It also decreases the risk of configuration errors [3].

The control plane houses the systems used to configure and control the emulated environments and the scenarios that run in the cyber range. In the second version of CRATE, the *CRATE Core API* (subsection 3.5) is used as the primary control channel between the cyber range operators and systems in the event plane. *CRATE Exercise Control (CEC)* (subsection 4.3) is a tool used to set up and manage training sessions. *SVED (Scanning, Vulnerabilities, Exploits and Detection)* (subsection 4.4) is a tool used to automate experiments and training scenarios.

Depending on the purpose of the cyber range event, the network data in the event plane may be collected along with relevant log files from the virtual machines or hardware devices. The *data collection* capability allows research on events such as training sessions, as exemplified in [18].

3.3 Emulated environments

The emulated environments in CRATE are set up as organisations. Each organisation contains at least one emulated network and each network contains one or more virtual machines and/or hardware devices. The virtual machines are created using templates defined in the CRATE Core API database.

CRATE is able to run several emulated environments in parallel without them affecting each other. There is no fixed limitation to the number of environments that can be run simultaneously, as this depends on the size and complexity of the organisations being emulated.

Normally, most emulated environments are not operational in the cyber range, but stored as definitions in a configuration database. The database contains more than one hundred different environments that are ready to be deployed in the cyber range. Some of these environments are used to create a *simulated internet*, that contains internet services such as backbone routers, DNS, RIPE Database, search engines and different web services such as social media and newspapers. The simulated internet enables realistic scenarios to be created in the cyber range.

A tool to automatically generate templates for emulated environments by setting some seed parameters is currently in development, and has been successfully tested in CRATE. This tool has the potential to help save time when creating large and complex environments for research and exercises.

3.4 Hardware devices

One key capability of CRATE is the ability to connect any type of *hardware device* anywhere in the emulated environments. Even though this capability may be used to conduct experiments with hardware-based security solutions, it is mainly used to build replicas of critical infrastructure with industrial control systems (ICS) and SCADA environments, as exemplified by [2]. CRATE hosts replicas of several critical infrastructure environments, including energy production and distribution, a traffic intersection, a railroad, an energy company, and a water purification plant. Several of these environments interact with the physical scale model called CRATE City.

However, using hardware devices make it costly to emulate larger environments due to device cost and the time needed to configure the systems. Few cyber ranges use virtualized industrial control systems as identified by Holm et al. [23]. None of the 30 testbeds included in the survey use virtualized industrial control systems or utilize automated tools to setup or control hardware devices.

To be able to create large and complex ICS environments, CRATE makes use of software based PLCs. The PLCs are based upon a modified version of OpenPLC [39]. One example of an emulated environment that makes use of this capability is the railroad system in CRATE City, which incorporates more than 70 software-based PLCs.

3.5 CRATE Core API

One of the lessons learned while operating the first generation of CRATE was that the cyber range needs to support continuous development to meet new requirements. Therefore, the second generation is centered around a new API, called CRATE Core API. The API runs as a service and is the central hub that manages all communication between the cyber range infrastructure and the different applications, as depicted in Figure 2.

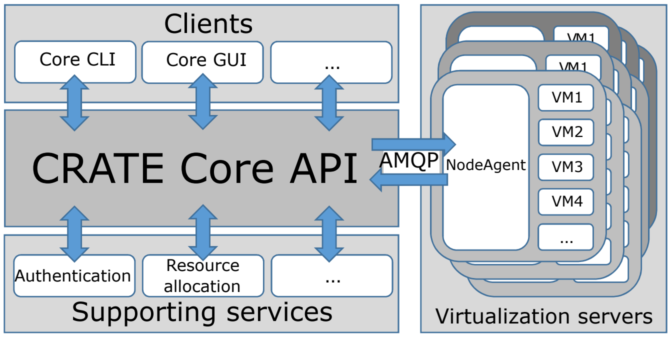

Fig. 2. A high level overview of the role of Core API

The API software is divided into three layers as shown in Figure 3. In the bottom layer, a database is used to store the configurations of the event environments to be emulated in CRATE. The server module of the API resides in the middle layer, and a series of API clients supply the user interfaces to the cyber range tools in the top layer [40].

CRATE Core API is built upon the Thrift framework [9]. One of the strengths with Thrift is that it can generate clients in several different programming languages [7]. Currently, clients are generated for Python and Java [40]. The Python client is primarily used for scripting purposes and the Java client is used by the graphical user interface called CRATE Core GUI. CRATE Core GUI is a web service that relies on the Vaadin framework [46]. Vaadin generates content in HTML and JavaScript dynamically from server code written in Java [29].

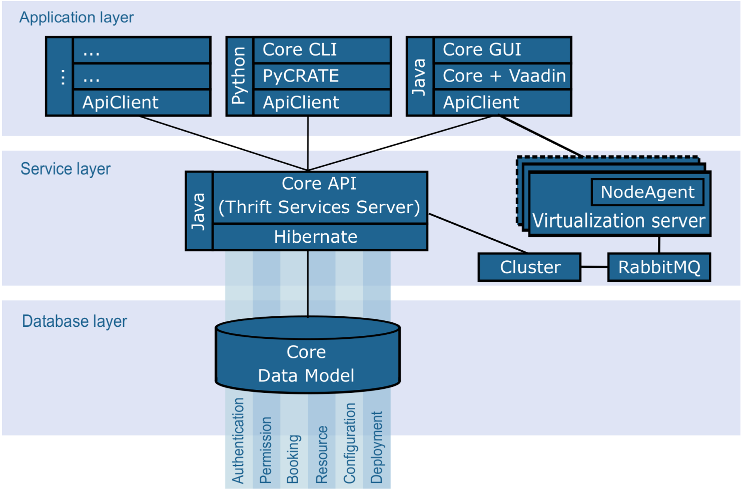

Fig. 3. The architecture of the Core API, showing the central role of the Thrift framework

Core API also provides several supporting services such as user authentication via LDAP and a service for cyber range resource reservations.

3.6 Cyber range users

The *cyber range users* are exemplified in Figure 1. Technical staff include the *developers* designing the cyber range tools and the *technicians* who operate the cyber range. The *tutors* use the cyber range for training sessions and exercises, and the *researchers* to conduct experiments.

During training sessions and exercises, users are often assigned to a team designated by a color matching their role, depicted in the top left corner in Figure 1. The most frequently used colors are blue representing the team being trained or the defending team, red representing counterplay or attacking team, white representing exercise management, and green representing the team managing the technical infrastructure [50]. The *user access* (subsection 3.6) takes place via command line or graphical user interfaces (GUIs).

To provide remote access to the cyber range, two solutions are available. Individual users are able to connect via a client-based VPN solution based on OpenVPN [32]. There is also a hardware-based solution where VPN boxes are used to create a site-to-site VPN. The VPN box contains a 48-port network switch and can be remotely administered through a web-based management tool. Each physical switch port can be mapped to a VXLAN (virtual network segment) representing an emulated network in an emulated environment. The automated configuration process takes less time than a manual operation and ensures that the configurations written to the devices are correct [3].

4 Automation features in CRATE

In the following subsections, we describe where and how the cyber range CRATE makes use of automation features.

4.1 Range provisioning

Range provisioning includes the preparation of the virtualization servers and configuration of the software-defined networks, processes that are fully automated in CRATE.

When virtual machines are to be deployed on a virtualization server, the NodeAgent service will configure the server's network with the required VXLANs (virtual network segments) and connect them to the virtual machines' network cards.

The automated range provisioning features also include the ability to reset a virtualization server to ”factory state”, wipe individual virtual machines or to upgrade the operating system, CrateOS. For an OS upgrade, the server only needs to be rebooted. All virtual machines and other settings are persistent.

4.2 System and service configuration

The process of configuring virtual machines and their services is fully automated in CRATE, and handled by the NodeAgent service. NodeAgent is a manager that runs on every virtualization server and it is subscribed to an AMQP queue, from where it reads instructions sent from the Core API. When NodeAgent receives the *deploy* command together with a JSON blob containing the virtual machine's configuration parameters, it will copy a virtual machine template to the server, start it and then run configuration commands through the hypervisor's API. NodeAgent will set configuration parameters such as hostname, local users and network settings. Services like DNS, Firewalls, gateways, directory servers and email servers are also configured. To ensure that a deployment was successful, NodeAgent will run a series of validation commands, and report the result back to Core API.

4.3 Exercise management

CRATE Exercise Control (CEC) is a web-based exercise management and support tool integrated into CRATE [6]. It is used to create and control scenarios, enhance the situational awareness during event execution, and score and evaluate performance of the participants after the training session or exercise. An event is created using a planning view, where injects are chosen from an inject database and scheduled on an event timeline. The database contain information about the inject, information on how the blue team may detect and report the inject, how the response should be scored by the white team and instructions for the red team on how to execute attacks when the inject is played. When exercise planning is ready, CEC generates a timeline view that can be used as a scheduler during the event.

During the event execution, red team activities can be automatically scheduled and launched from CEC if SVED is used. Scoring is also performed by scoring bots that monitor system and service availability in the event environment. CEC incorporates a view where incidents are reported and managed. Each report is also associated to an inject, which will enhance the situational awareness for the white team during the event.

The foundation for the after-action analysis consists of the event view where the reports are plotted chronologically, the scoreboard view, and information from the inject database [6].

4.4 Inject and test execution

SVED (Scanning, Vulnerabilities, Exploits and Detection) provides CRATE with a tool where actions are executed automatically and verbosely logged during an experiment or training session [21]. SVED increases the fidelity of the experiments executed in the cyber range by allowing actions to be executed in a reliable and repeatable manner. It also reduces the effort needed to run training sessions and exercises since the red team actions can be automated.

SVED consists of five components as described in [21]. A *threat intelligence* module collects system and vulnerability data from several different sources, including Core API, U.S. National Vulnerability Database and automatic scans performed with OpenVAS. A *designer* is used to create attack graphs via a GUI or via a script-based REST API. The *executioner* executes the attack graph and *attacker/sensor agents* runs commands or reports alerts in the emulated environments. The latter may be placed on any emulated network in the event plane, enabling SVED to mimic multiple attack patterns. Lastly, a *logger* stores log data generated when executing the attacks in the attack graph.

4.5 User emulation and traffic generation

The ability to emulate realistic user behavior and to generate network traffic is an essential part of a cyber range to enable realistic and relevant experiments, training sessions and exercises.

In CRATE, there are three methods used to automate user behavior. The first option is a bot that runs on the virtualization server and that uses the hypervisor's API to send instructions to the virtual machines. This option works best for command-line actions. The second option relies on the software AutoIt [10] and is used to automate software with graphical user interfaces, such as email clients and web browsers. The third option is integrated in the attack orchestration tool SVED, and is used where user actions are part of an attack. To emulate user behavior, SVED contains several pre-defined user actions that can be invoked, including sending and reading emails, opening files and attachments and visiting web pages. Different user behavior, for example risk-aware users or uneducated users, can be simulated with SVED by setting probabilities on the different user actions such as opening email attachments and clicking on links.

Traffic generation in CRATE relies on the traffic generated by user actions from the methods mentioned above.

4.6 Data collection

Data is usually collected from several sources during an event in CRATE, but only the traffic monitoring and intrusion detection system has yet been fully automated. Traffic monitoring and intrusion detection is done with system called SNART. SNART consists of several components that are configured to work together: a configuration component in CRATE Core GUI, an infrastructure component to collect network traffic from the network cards of the virtual machines and a dedicated virtual machine running TCPDump and Snort with the web GUI Snorby. The SNART system is configured in CRATE Core GUI and automatically deployed in the event environment.

5 Usage of automation in CRATE

The automated tools in CRATE are frequently used in the cyber range. In this section, we will exemplify how the automation enables or facilitates research experiments, training sessions and exercises.

5.1 Research experiments

Holm and Sommestad describe a experiment where SVED is used to investigate if the availability of offensive cyber tools decrease the skill required by an attacker to compromise a system [23]. During the research, SVED was used to automatically execute 1,223 exploits from 45 different exploit modules against 204 virtual machines in the cyber range. Without automation, this experiment would probably have been too labour-intensive to be possible.

[28] describes research performed in an emulated environment hosted in CRATE, where a generated scenario was executed automatically in a SCADA environment. The resulting dataset can also be used for future research. The environment used to perform the experiment is further described in [2].

In [26], Karresand, Axelsson, and Dyrkolbotn describe NTFS cluster allocation behavior. The experiments carried out during the research utilized automated capabilities in CRATE, including the creation of emulated environments as described in subsection 3.3 and 3.5 and the management of CrateOS described in subsection 4.1.

5.2 Training sessions and exercise events

In [42] and [47], two cyber security exercises in CRATE are described. During both events, CRATE Exercise Control (CEC) was used to automate exercise management and after-action analysis and evaluation, the latter enhancing the learning process of the blue teams. CEC also enabled the situational awareness during the exercises to be achieved without requiring a dedicated observer (often referred to as the yellow team). During repeated exercises and courses run in CRATE, CEC has proven capable of providing a good situational awareness during the events, making this task less labour-intensive, as described in subsection 4.3.

Another tool used to automate training sessions and exercises is SVED. During SAFE Cyber [47], SVED was used to perform the tasks normally performed by a red team by executing pre-configured attack graphs.

6 Automated tools in cyber ranges

As described in Section 2, numerous cyber ranges incorporate automated tools. However, the terminology used to describe the tools varies and the details available about the tools are sometimes scarce. Table 1 contains a compilation of the automated tools identified in the eleven cyber ranges as described in Section 2, 3 and 4. To facilitate comparison, the tools have been grouped into categories as described below.

Table 1. Automated tools used in eleven different cyber ranges.

	AIT CR	CCDCOE	CCE	CRATE	CyTRONE	DETERLab	Emulab	KYPO	LARIAT	U.S. NCR	VSCTC
Range setup											
Range provisioning				X	X	X	X	X	X	X	
Environment setup											
System deployment	X		X	X	X		X	X	X	X	X
System configuration	X		X	X	X		X	X	X	X	
Service configuration				X	X		X	X	X	X	
Hardware configuration											
Environment validation			X	X			X			X	
Event execution											
Environment adaptation			X								
Situational awareness		X	X	X					X		
Traffic generation							X			X	
User emulation				X					X	X	
Inject execution	X		X	X	X			X	X		
Test execution				X						X	X
Performance assessment											
System availability		X	X	X				X	X		
Service availability		X	X	X				X	X		
Data analysis			X								
Post-event actions											
Data collection										X	
System restore			X	X			X			X	

Range provisioning includes tools used to assign and setup cyber range infrastructure. *System deployment* refers to tools used to deploy pre-prepared virtual machines to create emulated environments. *System configuration* and *Service configuration* are used to setup and configure the virtual machines as well as their applications and services. *Hardware configuration* refers to tools used to setup and control hardware devices in the event plane, a capability none of the analyzed cyber ranges currently possess. Once deployed to the cyber range, the emulated environment is tested to ensure that is fulfills the defined requirements with *Environment validation* tools.

Environment adaptation includes tools used to change the environment during event execution and *Situational awareness* include automated tools that provide an overview and visualization of an event. *Traffic generation* refers to tools used to generate traffic in the emulated environments and *User emulation* includes tools used to mimic user behavior on the virtual machines. *Inject execution* is mainly used during training sessions and exercises, and includes automated execution of attacks. *Test execution* is focused on the execution of research experiments and tests in the emulated environments.

Automated performance assessment is mainly used during training sessions and exercises and includes measuring *System availability* as well as *Service availability.* The latter includes more advanced features such as synthetic logon and verifying service functionality. *Data analysis* encompasses tools used to derive the performance assessment based on data produced by the participants or their actions, such as logs or incident reports.

Post-event actions conclude the table. *Data collection* refers to tools that automate the collection of data from multiple sources after an event. *System restore* includes tools used both to release assigned cyber range infrastructure, reset the emulated environments and, when needed, completely erase the event data to prevent data leakage.

Note that Table 1 only includes tools used to automate the cyber range itself and not tools used within the emulated environments, such as scanning tools or analytic tools. Nor does the table include generic IT tools that are manually configured to perform a task, such as sniffers, scanners or monitoring tools.

Figure 4 displays the number of automated tools identified in the eleven cyber ranges included in Table 1.

All but one of the analyzed cyber ranges include automated tools to setup and control emulated environments. During the event execution, a majority of the analyzed cyber ranges utilize automated tools to execute injects. Automated tools used to assess performance is included in five cyber ranges, four of which also include automated tools to enhance the situational awareness. All of these cyber ranges are described as used for training sessions and exercises.

7 Discussion

The information available about the automated tools in cyber ranges has proven to be rather limited. The automation features are often mentioned only in a few

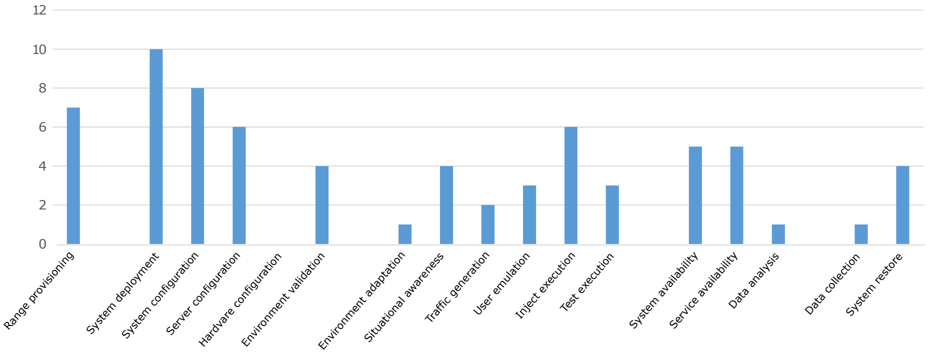

Fig. 4. Number of automated tools in the eleven cyber ranges included in Table 1

sentences, and it is hard to assess the maturity or extent of a certain tool, or even if it is operational or just an identified requirement. The terminology used to describe the tools varies between different papers, and when an evaluation of the automated tools are included, they are normally only compared to performing the same task manually in the same cyber range. All together, these circumstances makes it hard to compare tools in different cyber ranges.

The tools in Table 1 are included based on the assessments that could be made based on information available. It is therefore quite possible that tools are incorrectly included, or left out of Table 1. Furthermore, the data in Table 1 should not be seen as a comparison of cyber range capabilities, since the data available is too limited to perform such a comparison.

Two of the surveys, [51] and [25], used as sources in this paper identifies an automation trend in cyber ranges starting around 2014. However, our findings indicate that many cyber ranges have been using automated tools to setup and control emulated environments several years prior to 2014. Even though the reason for this deviation has not been exhaustively analyzed while writing this paper, our theory is that it depends on how cyber range automation tools are described in research papers.

8 Conclusions

In this paper we have presented a compilation of automated tools used in cyber ranges, as well as several research initiatives designated to further increase cyber range automation. We have also presented the cyber range CRATE, operated by the Swedish Defence Research Agency, and described its automation features. We have found that automated tools have been used to setup and control emulated environments in cyber ranges for several years, and that many cyber ranges include such tools today. We have also identified that there is a need to further use automation to be able to increase the number of cyber range events and to increase the fidelity of the experiments executed.

References

1. Abbott, R., Mcclain, J., Anderson, B., Nauer, K., Silva, A., Forsythe, C.: Automated performance assessment in cyber training exercises. In: Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) (2015)
2. Almgren, M., Andersson, P., Björkman, G., Ekstedt, M., Hallberg, J., Nadjm-Tehrani, S., Westring, E.: Rics-el: Building a national testbed for research and training on scada security (short paper). In: Luiijf, E., Žutautaitė, I., Hämmerli, B.M. (eds.) Critical Information Infrastructures Security. pp. 219–225. Springer International Publishing, Cham (2019)
3. Almroth, J.: Mjukvarudefinierade nätverk i CRATE. Tech. Rep. FOI Memo 6386, The Swedish Defence Research Agency (2018)
4. Almroth, J.: Design-, krav- och funktionsspecifikation CRATE 2.0. Tech. Rep. FOI Memo 6666, The Swedish Defence Research Agency (2019)
5. Almroth, J.: Nationell Cyber Range CRATE 2.0 - Virtualiseringsnoder 2.0. Tech. Rep. FOI Memo 6710, The Swedish Defence Research Agency (2019)
6. Almroth, J., Gustafsson, T.: Crate exercise control – a cyber defense exercise management and support tool. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroSPW) (2020)
7. Almroth, J., Härje, T.: Dokumenterat utvecklingsramverk för CRATE 2.0. Tech. Rep. FOI Memo 6381, The Swedish Defence Research Agency (2018)
8. Anderson, D.S., Hibler, M., Stoller, L., Stack, T., Lepreau, J.: Automatic online validation of network configuration in the emulab network testbed. In: 2006 IEEE International Conference on Autonomic Computing. pp. 134–142 (2006). https://doi.org/10.1109/ICAC.2006.1662391
9. Apache Software Foundation: Thrift framework homepage, `https://thrift.apache.org/`, last accessed on 9 August 2020
10. AutoIt Consulting Ltd.: Autoit homepage, `https://www.autoitscript.com/`, last accessed on 23 August 2020
11. Braje, T.: Advanced tools for cyber ranges. Lincoln laboratory journal **22**(1) (2016)
12. Burke, I., van Heerden, R.: Automating cyber offensive operations for cyber challenges (03 2016)
13. Costa, G., Russo, E., Armando, A.: Automating the generation of cyber range virtual scenarios with vsdl. ArXiv (2020)
14. Davis, J., Magrath, S.: A survey of cyber ranges and testbeds. Tech. rep., Defence science and technology organisation Edinburgh (Australia) Cyber and electronic warfare division (2013)
15. Ferguson, B., Tall, A., Olsen, D.: National cyber range overview. In: 2014 IEEE Military Communications Conference. pp. 123–128 (2014)
16. Frank, M., Leitner, M., Pahi, T.: Design considerations for cyber security testbeds: A case study on a cyber security testbed for education. In: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress. pp. 38–46 (2017). https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.23
17. Gustafsson, T.: The crate network generator - an automated method for building the dynamic and scalable network architecture of a cyber-range. In: Proceedings: SNCNW 2013 - 9th Swedish National Computer Networking Workshop (2013)
18. Hammervik, M., Granåsen, D., Hallberg, J.: Capturing a cyber defence exercise. In: TAMSEC - Technology and Methodology for Security and Crisis Management (2010)

19. Hibler, M., Ricci, R., Stoller, L., Duerig, J., Guruprasad, S., Stack, T., Webb, K., Lepreau, J.: Large-scale virtualization in the emulab network testbed. pp. 113–128 (01 2008)
20. Holm, H.: Baltic cyber shield - research from a red team versus blue team exercise. PenTest Magazine pp. 80–86 (05 2012)
21. Holm, H., Sommestad, T.: Sved: Scanning, vulnerabilities, exploits and detection. In: MILCOM 2016 - 2016 IEEE Military Communications Conference. pp. 976–981 (11 2016). https://doi.org/10.1109/MILCOM.2016.7795457
22. Holm, H., Karresand, M., Vidström, A., Westring, E.: A survey of industrial control system testbeds. In: Buchegger, S., Dam, M. (eds.) Secure IT Systems. pp. 11–26. Springer International Publishing, Cham (2015)
23. Holm, H., Sommestad, T.: So long, and thanks for only using readily available scripts. Information and Computer Security **25**, 47–61 (03 2017). https://doi.org/10.1108/ICS-08-2016-0069
24. Joonsoo, K., Youngjae, M., Moonsu, J.: Becoming invisible hands of national live-fire attack-defense cyber exercise. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 77–84. IEEE (2019)
25. Karlzén, H.: Omvärldsstudie om cyberanläggningar. Tech. Rep. FOI Memo 7213, The Swedish Defence Research Agency (2020)
26. Karresand, M., Axelsson, S., Dyrkolbotn, G.: Using ntfs cluster allocation behavior to find the location of user data. Digital Investigation **29**, S51–S60 (2019). https://doi.org/https://doi.org/10.1016/j.diin.2019.04.018
27. Leitner, M., Frank, M., Hotwagner, W., Langner, G., Maurhart, O., Pahi, T., Reuter, L., Skopik, F., Smith, P., Warum, M.: Ait cyber range: Flexible cyber security environment for exercises, training and research. In: European Interdisciplinary Cybersecurity Conference (EICC) (2020), `https://pdfs.semanticscholar.org/d4b6/11aee8dcca086e4f473f76dfe996a61149cf.pdf?\_ga=2.223971871.580539717.1603710325-1968247500.1591902000`, to be presented at EICC'20, November 18-19
28. Lin, C.Y.: A timing approach to network-based anomaly detection for SCADA systems (06 2020). https://doi.org/10.3384/lic.diva-165155
29. Lundholm, K.: Nationell cyber range CRATE 2.0 - CRATE CORE och CRATE GUI. Tech. Rep. FOI Memo 6711, The Swedish Defence Research Agency (2019)
30. Melón, F., Väisänen, T., Pihelgas, M.: Eve and adam: Situation awareness tools for nato ccdcoe cyber exercises. In: Systems Concepts and Integration (SCI) Panel SCI-300 Specialists' Meeting on 'Cyber Physical Security of Defense Systems' (2018)
31. Neville, S., Li, K.: The rational for developing larger-scale 1000+ machine emulation-based research test beds. pp. 1092–1099 (2009). https://doi.org/10.1109/WAINA.2009.183
32. OpenVPN Inc.: Openvpn homepage, `https://www.openvpn.net/`, last accessed on 19 August 2020
33. Pham, C., Tang, D., Chinen, K.i., Beuran, R.: Cyris: a cyber range instantiation system for facilitating security training. pp. 251–258 (12 2016). https://doi.org/10.1145/3011077.3011087
34. Pihelgas, M.: Design and implementation of an availability scoring system for cyber defence exercises. Proceedings of the 14th International Conference on Cyber Warfare and Security pp. 329–337 (2019)
35. Pridmore, L., Lardieri, P., Hollister, R.: National cyber range (ncr) automated test tools: Implications and application to network-centric support tools. In: 2010 IEEE AUTOTESTCON. pp. 1–4 (2010)

36. Russo, E., Costa, G., Armando, A.: Scenario design and validation for next generation cyber ranges. In: 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). pp. 1–4 (2018)
37. Russo, E., Costa, G., Armando, A.: Building next generation cyber ranges with crack. Computers Security **Volume 95** (04 2020). https://doi.org/10.1016/j.cose.2020.101837
38. Shu, G., Chen, D., Liu, Z., Li, N., Sang, L., Lee, D.: Vcstc: Virtual cyber security testing capability - an application oriented paradigm for network infrastructure protection. pp. 119–134 (01 2008). https://doi.org/10.1007/978-3-540-68524-1_10
39. Sklower, K., Joseph, A.D.: Very large scale cooperative experiments in emulab-derived systems. In: Proceedings of the DETER Community Workshop on Cyber Security Experimentation and Test. DETER, USENIX Association, USA (2007)
40. Sohlmér, M.: Utveckling av crate core api och crate core gui. Tech. Rep. FOI Memo 6885, The Swedish Defence Research Agency (2019)
41. Sommestad, T.: Experimentation on operational cyber security in crate. Tech. rep., NATO STO-MP-IST-133 Specialist meeting (2017)
42. Strålskyddsmyndigheten: It-attack mot kärntekniska anläggningar övas, `https://www.stralsakerhetsmyndigheten.se/press/nyheter/2017/it-attack-mot-karntekniska-anlaggningar-ovas/`, last accessed on 26 May 2020
43. Tang, D., Pham, C., Chinen, K., Beuran, R.: Interactive cybersecurity defense training inspired by web-based learning theory. In: 2017 IEEE 9th International Conference on Engineering Education (ICEED). pp. 90–95 (2017)
44. The Deter Project: Deterlab capabilities, `{https://deter-project.org/deterlab\_capabilities}`, last accessed on 21 October 2020
45. Urias, V.E., Stout, W.M.S., Van Leeuwen, B., Lin, H.: Cyber range infrastructure limitations and needs of tomorrow: A position paper. In: 2018 International Carnahan Conference on Security Technology (ICCST). pp. 1–5 (2018)
46. Vaadin Ltd.: Vaadin homepage, `https://vaadin.com/`, last accessed on 19 August 2020
47. Valassi, C., Wedlin, M.: Övningsrapport: Safe cyber 2019. planering, utveckling, genomförande och lärdomar av en storskalig cdx-övning. Tech. Rep. FOI-R–4885–SE, The Swedish Defence Research Agency (2020)
48. Vykopal, J., Oslejsek, R., Čeleda, P., Vizváry, M., Tovarňák, D.: Kypo cyber range: Design and use cases. In: ICSOFT (2017)
49. White, B., Lepreau, J., Stoller, L., Ricci, R., Guruprasad, S., Newbold, M., Hibler, M., Barb, C., Joglekar, A.: An integrated experimental environment for distributed systems and networks. SIGOPS Oper. Syst. Rev. **36**(SI), 255–270 (Dec 2003). https://doi.org/10.1145/844128.844152
50. Wilhelmson, H., Svensson, T.: Handbook for planning, running and evaluating information technology and cyber security exercises. Handbook, Center for Asymmetric Threats Studies, Swedish Defence University (2014)
51. Yamin, M., Katt, B., Gkioulos, V.: Cyber ranges and security testbeds: Scenarios, functions, tools and architecture. vol. 88 (10 2019). https://doi.org/10.1016/j.cose.2019.101636
52. Yasuda, S., Miura, R., Satoshi, O., Takano, Y., Miyachi, T.: Alfons: A mimetic network environment construction system. pp. 59–69 (06 2016). https://doi.org/10.1007/978-3-319-49580-4_6