FURTHER OPTIMISING COMPUTER SECURITY ANALYSIS BY USING MORE EFFICIENT AUTOMATED ADVERSARY EMULATION TOOLS

Fawwaz Ahamed[1], Lim Seh Leng[2]
[1]Victoria School, 2 Siglap Link, Singapore 448880
[2]Defence Science & Technology Agency, 1 Depot Road. Singapore 109679

Abstract:

This research aims to accomplish the following:

1. Compare the efficiency of the tools by analysing the "Noise" produced by each tool
2. Compare the Extensibility, Versatility and Ease-of-use of each tool
3. Look into how to advance the current state of Computer Security using the tools tested

Contents:

1. Introduction

Importance of Computer Security

In our world today, mostly all complex systems, from Health-related Devices to Government Databases, rely extensively on Computers, due to their versatility, speed of task completion and the huge efficiency benefits that they bring. It is an understatement to say that these computers are essential and well-rooted to our society and to our world. To put it into context, just like how Oil was to the world economy the previous decade, Computers are to the world this decade, and maybe beyond.

With such importance and space given to computers in our society, we cannot understate the importance of security that we have to enforce to secure and protect these vital and crucial computer systems, which just so happens to affect our lives directly.

Introduction to Red Teaming

Red Team is a group that plays the role of an enemy or competitor to provide security feedback from that perspective. This allows us to identify and flag flaws and holes in computer systems that can be patched / corrected to enforce the security of the system. This prevents valuable data from being stolen / corrupted / forged / modified by unauthorised personnel.

Red Teaming commonly involves cybersecurity research with the use of role-playing adversaries. Red Team analysts emulate these adversaries on the computer system that they need to test and monitor the activity and the outcomes of the test. This tells them whether the computer system is resilient to such attacks. After running the emulation tests, analysts review the logs that were created during the emulation exercise in order to understand the faults in the System, which is later corrected / patched.

However, to train analysts to perform such tests manually takes a lot of time and money. An Automated Adversary Emulation Tools would allow us to mitigate the costs of training since the series of tests are conducted automatically by scripts using a certain framework, e.g. MITRE ATT&CK.

Characteristics of the Automated Adversary Emulation Tools

There exists many Automated Adversary Emulation Tools. However, each tool is unique in the methods used to run them and scripts that are ran in the background in order to execute the adversary emulations and the efficiency of the programs & codes behind the Automated Adversary Emulation Tools. Hence, when emulation exercises are ran on the "Victim Machine", more often than not, along with leaving logs that were created by the emulation itself, the "Victim Machine" also records log entries for the internal processes carried out by the Automated Adversary Emulation Tools. Hence, the results of the emulation exercise, i.e. the recorded logs during the period of adversary emulation, is "contaminated" by the unwanted logs, or "noise", that the Automated Emulation tools produce.

In order to identify the Automated Adversary Emulation Tool that is the most efficient and effective on the basis of the least amount of “noise” produced, this research has been conducted. This research involved running the same set of adversaries being emulated using 2 Automated Adversary Emulation Tools, MITRE Caldera and Red Canary Invoke-Atomic Red (also known as Atomic Red)

At the end of the research, this research paper aims to identify the particular reasonings behind the difference in the number of logs that each Automated Adversary Emulation Tool produces after each emulation exercise.

As an extension to the project, this research paper also aims to test the nature of the tools and assess on their characteristics in the real-world setting, i.e. how well do the tools adapt to the real world. Moreover, this research would be using a new tool, PurpleSharp, to enhance and elaborate on the findings. In addition, this research would focus on the Blue Teaming side, whereby assessing whether the Automated Red Teaming tools assessed in the above tests could also be used for a small-scale deployment of Blue Team.

2. Materials and Method:

2.a Materials:

A few materials were used to conduct the tests necessary for this research, listed below:

Hardware / Software Information:

Hardware / Software	Specification
2.a.i Host Machine	
CPU	AMD Ryzen 7 5800H, 8 Cores 16 Threads, 3.20 GHz
Memory, Storage	16.0 GB, 512.0 GB (SSD)
Operating System	Windows 11 Home 22H2 Build: 22621.607
	Windows Feature Experience Pack 1000.22634.1000.0, AMD64
2.a.ii Oracle VirtualBox	
Version	6.1.30 r148432 (Qt5.6.2)
2.a.iii Linux Virtual Machine (Server for Caldera):	
CPU	AMD Ryzen 7 5800H, 2 Cores 2 Threads, 3.20 GHz
Memory, Storage	2.0 GB, 35.0 GB (VDI)
Operating System	Ubuntu 22.04.1 LTS, x86_64
2.a.iv Mitre Caldera	
Version	4.1.0
2.a.v Red Canary Invoke-Atomic Red (also known as Atomic Red)	
Version	1.0.2
2.a.vi Victim Virtual Machine	
CPU	AMD Ryzen 7 5800H, 4 Cores 4 Threads, 3.19 GHz – 3.95GHz
Memory, Storage	4.0 GB, 50.0 GB (VDI)
Operating System	Windows 10 Pro 21H1, Build: 19043.1949
	Windows Feature Experience Pack 120.2212.4180.0, x64

Table 2.1: Information on Hardware/Software used

2.b Method

An advanced persistent threat (APT) is a broad term used to describe an attack campaign in which an intruder, or team of intruders, establishes an illicit, long-term presence on a network in order to mine highly sensitive data [1]. There exist numerous groups of APTs. For the test conducted, abilities from across a few Advanced Persistent Threat (APT) groups were selected.

Abilities are the building blocks for the operations that are carried out. Abilities are predefined actions that would be executed at the victim machine. Such abilities are classified into respective techniques, governed by a certain framework, in the case of APT, the MITRE ATT&CK Framework.

Since Caldera only supports abilities under the APT3 group out of the box, a native plugin named "Atomics" was required to be enabled to import other abilities from other APTs into Caldera. A few abilities under the above mentioned APTs have been cherry-picked for this test on the basis of whether both emulation tools are able to run these abilities on the victim machine without errors, the combination of these test would leave a notable trace on the victim machine so as to detect for noise and assorting the range of attacks, from Discovery to Collection, with differently weighed attacks in each category, from the easier ones such as copy clipboard to the heavier ones such as take screenshot and store on victim machine's desktop. The list of abilities used is as follows:

2.b.i Abilities Used

Ability ID	**Name of Ability**	**Category**	**Description of Ability**
T1113	Screen Capture	Collection	Capture's screen and saves to Victim's Desktop
T1057	System Processes	Discovery	Discovers System Processes
T1087.001	Identify Local Users	Discovery	Account Discovery: Local Account
T1124	Get System Time	Discovery	System Time Discovery
T1115	Copy Clipboard	Collection	Gets Clipboard Data
T1078.001	Activate Guest Account	Multiple	Activates Default Guest Account
T1518	Applications Installed	Discovery	Queries registry for the list of applications installed and their versions
T1531	Change User Password	Impact	Changes a User's Password
T1218.002	Control Panel Items	Defence-Evasion	Simulates an adversary that leverages on control.exe. Upon execution, calc.exe is launched
T1078.003	Create local Admin Account	Multiple	A new Admin account would be created

Table 2.2: List of Abilities used for Experiment

2.b.ii Data Collection Mechanism

After running the defined test, shown above, Windows Event Viewer was referred to get Windows Logs. In order to classify the logs into distinct operations for easier evaluation after experiment, System Monitor (Sysmon) was used, which is installed onto the "Victim Machine" as an Administrator. Sysmon is a Windows system service and device driver that, once installed on a system, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creations, network connections, and changes to file creation time [2].

2.b.iii Flow of Experiment

Each tool makes use of different techniques and methods to deploy the tests. Each method is described below:

Before Running Tests:

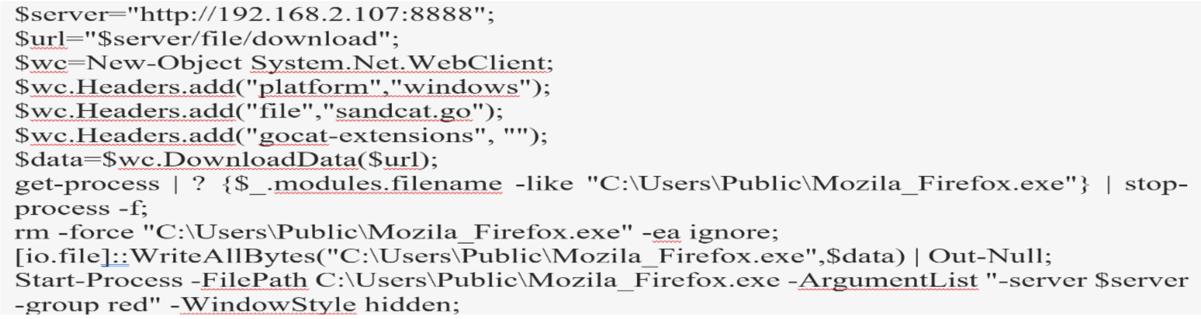
Before initiating any test, the Event Viewer is cleared using the following Command Prompt command ran as an Administrator on the Victim Virtual Machine:

```
for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1"
```

This is to allow us to only have the logs that are generated from the test run at concern so as to prevent errors during recording the results of each test. The subsequent tests are ran as a Non-Administrator User and with Windows Defender and Firewall turned off.

Running the test on Caldera:

After clearing the Windows Event Viewer Logs, the Caldera Control Webpage was accessed on the Victim Machine. Following which, an agent deployment PowerShell Script was generated. The script is shown below:



```
$server="http://192.168.2.107:8888";
$url="$server/file/download";
$wc=New-Object System.Net.WebClient;
$wc.Headers.add("platform","windows");
$wc.Headers.add("file","sandcat.go");
$wc.Headers.add("gocat-extensions", "");
$data=$wc.DownloadData($url);
get-process | ? {$_.modules.filename -like "C:\Users\Public\Mozila_Firefox.exe"} | stop-
process -f;
rm -force "C:\Users\Public\Mozila_Firefox.exe" -ea ignore;
[io.file]::WriteAllBytes("C:\Users\Public\Mozila_Firefox.exe",$data) | Out-Null;
Start-Process -FilePath C:\Users\Public\Mozila_Firefox.exe -ArgumentList "-server $server
-group red" -WindowStyle hidden;
```

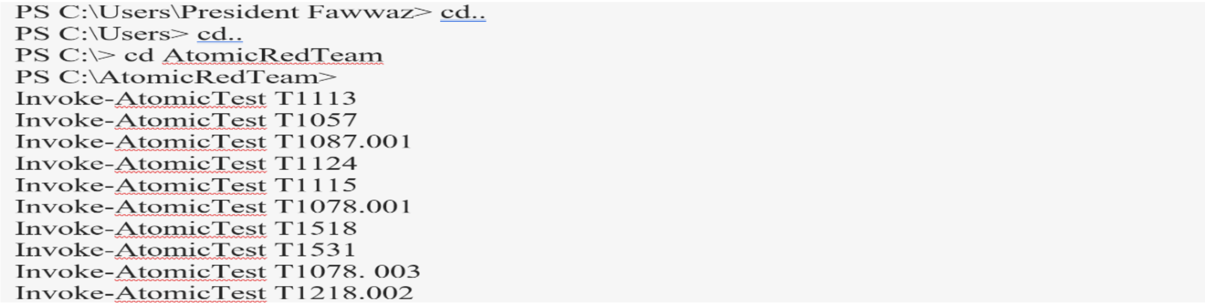
This PowerShell Script was run on the Windows PowerShell ISE on the Victim's Machine in order to deploy an agent, Mozila_Firefox.exe, onto the Victim Machine.

After having planted the agent into the Victim Machine, the above seen tests are run on Caldera, After running these tests, data is collected from the Windows Event Viewer Sysmon folder under the following directory:

```
Applications and Services Logs/Microsoft/Windows/Sysmon/Operational
```

Running the test on Red Canary Invoke-Atomic Red:

After clearing the Windows Event Viewer Logs, the PowerShell ISE is accessed on the Victim Machine. After which, directory is changed to the AtomicRedTeam directory, usually in the root of the C Drive. Following which, the script used to run the test shown above is key in. The PowerShell operation is shown below:



```
PS C:\Users\President Fawwaz> cd..
PS C:\Users> cd..
PS C:\> cd AtomicRedTeam
PS C:\AtomicRedTeam>
Invoke-AtomicTest T1113
Invoke-AtomicTest T1057
Invoke-AtomicTest T1087.001
Invoke-AtomicTest T1124
Invoke-AtomicTest T1115
Invoke-AtomicTest T1078.001
Invoke-AtomicTest T1518
Invoke-AtomicTest T1531
Invoke-AtomicTest T1078. 003
Invoke-AtomicTest T1218.002
```

After entering the PowerShell Script into the PowerShell ISE on the Victim Machine, the above seen tests are run on Invoke-Atomic Red, After running these tests, data is collected from the Windows Event Viewer Sysmon folder under the following directory:

Applications and Services Logs/Microsoft/Windows/Sysmon/Operational

3. Results:

3.a Results from Test

Sysmon Event ID	Caldera	Invoke-Atomic Red
1 - Process Creation	40	114
5 - Process Terminated	0	0
11 - File Create	8	18
13 - Registry Event (Value Set)	14	0
22 - DNS Event (DNS Query)	0	6
Total Logs Generated	70	145
Logs after Filtering	64	144
Logs Filtered Away (Logs due to other System Processes)	6	1

Table 3.1: Results from Experiment

3.b Comparison and Conclusion from the Experimental Data

Results recorded from the experiment displays that when ran identical set of tests of the automated emulation tools at concern, Mitre Caldera and Red Canary Invoke-Atomic Red, Mitre Caldera runs the set of tests with generating lesser logs when compared to Red Canary Invoke-Atomic Red. This shows us that Caldera performs APT Emulations on Victim Machines with a lower amount of "noise" in the System Logs when we compare it with Invoke-Atomic Red.

4. Discussion:

4.a Reason for such a difference

Even if the Adversary Emulated is ran using the same script, the contributing factor for this difference is the fact that either tool requires a different method of deployment, i.e. Caldera is deployed from a Linux server, while Invoke-Atomic Red is deployed locally on the Victim Windows Machine. The following case-study elaborates the above further.

Deploying Caldera:

A web interface is used to deploy Caldera. This web interface directly communicates to the Caldera Server on the Linux machine, which then disseminates the C2 Commands, i.e. Terminal Commands that trigger the intended action to take place on the Victim's Machine, to the target Victim Machine. Hence, all file queries, tool-related activities, internal logs recorded by the tools, modifications to the application data, i.e. the tool's internal data, are all done on the Linux Server. Hence, all these changes do not affect the Victim Windows Machine. Therefore, logs recorded in the Windows Event Viewer during the runtime of the emulation exercise are mainly due to running the C2 Commands themselves, making this deployment a more "noise-resistant" one.

Deploying Invoke-Atomic Red:

The PowerShell Terminal on the Victim Machine is used to deploy Invoke-Atomic Red. Hence, file query operations to search for the relevant ability scripts, tool-related activities and essential services of the tool would also run on the Victim Machine during the runtime of the emulation exercise. This would not only cause logs to be produced on the Windows Event Viewer by the C2 Commands ran during the emulation exercise, but also by other tool-related background activities. This would contaminate the logs recorded during the runtime of the emulation exercise, making this deployment a more "noisy" one.

The above case-studies are clear in the fact that Caldera is less "noisier" than Invoke-Atomic Red since it isolates the tool's background activity from the Victim Machine. This conclusion and discussion is in line with the results, which shows us that Caldera produces fewer logs than Invoke-Atomic Red.

5. Extension:

In this research, other aspects of these tools, such as the customisability and Blue Team reactions have also been assessed. Moreover, another tool has been tested.

5.a Nature of the Tools:

The tools at question have been assessed for their customisability, crucial for organisations that would like to expand Cybersecurity analytics further. For which, the following crucial aspects of customisability have been assessed.

5.a.i Extensibility

Both Mitre Caldera and Invoke-Atomic Red have similar characteristics in importing new abilities to emulate since both pool Ability Data, i.e. the script that runs the ability, from the same GitHub Server. Likewise, if analysts would like to import new abilities into the tools to test, both tools would be competent since both tools work on .yaml files, i.e. the .yaml files contain the program for the ability. Hence, new abilities written in this format would run smoothly on both tools, provided that the same directory structure is replicated.

5.a.ii Versatility

Both Mitre Caldera and Invoke-Atomic Red are equally versatile. This is because, both Caldera and Invoke-Atomic Red are tools that are used to run scripts, regardless of blue or red team. Moreover, both Caldera and Invoke-Atomic Red are open-sourced. Hence, analysts could tune, reshape and easily customise the tools to their preferences and needs.

5.a.iii Ease of Use

When testing both Mitre Caldera and Invoke-Atomic Red, Mitre Caldera is seemed to be much more user-friendly amongst the two. This is due to the useful visual classification of the abilities into different techniques, with coherence to the Mitre ATT&CK Framework. Moreover, a graphical UI allows the user to construct test sequences, deploy the tests and view the results more easily, when compared to Invoke-Atomic Red, which needs a PowerShell Terminal as its UI. Hence, as Caldera offers a more User-Friendly experience due to its graphical interface, which visualises the activities and aids the user in navigation and conducting the tests, Caldera has a less-steep learning curve for the users to overcome, making it easier to use when compared with Invoke-Atomic Red.

5.b PurpleSharp - A new tool

In order to verify the test results and conclusion, another tool, PurpleSharp, was ran. PurpleSharp is an open-source Automated Adversary Emulation Tool that is found on GitHub (https://github.com/mvelazc0/PurpleSharp). It only runs on Windows Machines, be it local or over TCP/IP, as it is written in C#, which requires the .net Framework by Microsoft to run it. Another test, with the same list of Abilities used from Table 2.2, was conducted locally on the Victim Windows Machine. The results of the test are below, which is compared with Mitre Caldera:

Sysmon Event ID	PurpleSharp (Local)	Caldera (TCP/IP)
1 - Process Creation	6	40
5 - Process Terminated	1	0
11 - File Create	8	8
13 - Registry Event (Value Set)	1	14
22 - DNS Event (DNS Query)	0	0
Total Logs Generated	16	70
Logs after Filtering	16	64

Table 5.1: Results from Experiment

The reason for PurpleSharp being a lot less “Noisier” than Caldera, i.e. generating a lot less logs than Caldera, is due to the fact that both PurpleSharp and Caldera run the Adversaries on the Victim Machine differently. Once the Caldera server sends the C2 command to the agent on the Victim Machine, the C2 Commands are run as individual processes, hence more Process Creation logs are recorded. However, when PurpleSharp was run locally, since PurpleSharp is compiled as an .exe file and written in C#, all commands and activities are ran inside the application. Hence, when running the list of abilities from Table 2.2, we observe only 6 process creation logs, out of which, 1 is due to the start-up of the PurpleSharp.exe application, and the other 5 is due to the processes that are demanded by the abilities ran. We could hence conclude that the method of emulating the adversaries has a significant impact on the test exercise.

5.c Blue Team Operations:

A Blue Team is a group of analysts who perform analysis on Information Systems, i.e. computers, to ensure security, identify security threats, verify effectiveness of each security measure. This is done by listening, or watching over, the system processes ran on the system, identifying & locating system processes which are a threat to the system, usually by looking out for the commands and programs ran by the process at question, and flagging it as a threat. Manually performing Blue Team tasks are really tedious and expensive since a suite of staff have to be given training and a lot of time in order to perform manual Blue Teaming. However, automated Blue Teaming solutions conduct Blue Teaming in a more efficient way since it requires minimal human-work to get the operation deployed and completed. An Automated Blue Team tool usually runs on a server on another machine, then plants an agent on the machine which needs to be analysed, and then listens out to all commands & programs ran and process active and then automatically flag threats.

The tools tested, i.e. Mitre Caldera, Invoke-Atomic Red Team and PurpleSharp are also assessed on their Blue Team abilities. This is important since it would be more efficient and cost-effective if these Automated Adversary Emulation tools have the ability to be a low-cost, lightweight, quick-fix solution to temporarily replace existing Blue Team solutions. Such temporary replacements are inevitable due to late software patch distribution by the solution provider, or handy for small tests, where a quick and light deployment is efficient. Using these Automated Red Team Tools for Blue Teaming is also efficient since both operations could be ran simultaneously from the same tool. The following would list the findings of this assessment:

Mitre Caldera

Mitre Caldera has an inbuilt Blue Teaming capability and some plugins catered towards Blue Teaming, such as Gameboard, alongside its Red Team abilities. This allows analysts to run both Red Team and Blue Team tests on the same platform. Deployment is as similar to Red Team deployment on Mitre Caldera.

Invoke-Atomic Red Team

Invoke-Atomic Red Team does not have an inbuilt Blue Teaming capability. Hence, it is unable to run Blue Team.

PurpleSharp

Despite the name, PurpleSharp does not have an inbuilt Blue Teaming capability. Hence, it is unable to run Blue Team.

6. Acknowledgements:

I would like to acknowledge Ms Lim Seh Leng, DSTA, my mentor for this research paper, for supporting and guiding me throughout my research. She has been very helpful guiding me about my challenges when I stumble upon any and was very encouraging, which

allowed me to push myself further than the required scope of this project. I certainly appreciate her mentorship and am grateful for being a research intern under her for 5 months.

7. References:

[1] - “What Is APT (Advanced Persistent Threat): APT Security: Imperva.” Learning Center, December 29, 2019. https://www.imperva.com/learn/application-security/apt-advanced-persistent-threat/.

[2] - Markruss. “Sysmon - Sysinternals.” Sysmon - Sysinternals | Microsoft Learn. Accessed December 4, 2022. https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon.