POSTER: Construct macOS Cyber Range for Red/Blue Teams

Yi-Hsien Chen[1,2], Yen-Da Lin[2], Chung-Kuan Chen[2], Chin-Laung Lei[1], Chun-Ying Huang[3]
[1]Department of Electrical Engineering, National Taiwan University
[2]CyCraft Technology Corporation
[3]Department of Computer Science, National Chiao Tung University
Email:csjh21010@gmail.com,segno.dada@gmail.com,ck.chen@cycraft.com,cllei@ntu.edu.tw,chuang@cs.nctu.edu.tw

ABSTRACT

More and more malicious apps and APT attacks now target on macOS, and it is therefore crucial for researchers to develop threat countermeasures on macOS. In this paper, we attempt to construct a macOS cyber range for evaluating red team and blue team performance. Our proposed system is composed of three fundamental components: an attack-defense association graph, a Go language-based red team emulation tool, and a toolkit for blue team performance evaluation. We demonstrate the effectiveness of our proposed cyber range with real-world scenarios, and believe it will stimulate more research innovations on threat analysis for macOS.

CCS CONCEPTS

• **Security and privacy** → *Intrusion/anomaly detection and malware mitigation*;

KEYWORDS

Forensic, Blue Team, Cyber Range, macOS Security, Red Team, Penetration Testing

1 INTRODUCTION

There is an increasing number of users using macOS and therefore more and more threat actors target on attacking macOS. For instance, state-sponsored APT28 utilized Trojan.MAC.APT2 to attack military and government organizations [1]. Malware examples such as OSX.AppleJeus [5] and OSX.NetWire.A [4] are widely used by malicious actors to attack cryptocurrency exchanges. Although macOS is popular, we observed that seldom research works discuss attack and defense techniques on macOS. As a result, both blue teams and red teams are not acquainted with macOS security techniques including attack methods, protection mechanisms and tools for investigating. Thus, a systematic survey of macOS attack and defense technique is demanded, and a modularized cyber range for training red teams and blue teams would greatly improve the skills and experiences of the teams.

In this paper, we attempt to resolve the aforementioned issues by building a cyber range for macOS. Figure 1 shows the architecture of our proposed cyber range. The cyber range is composed of three components. First, we propose building an attack-defense association graph, which systematically summarizes possible attack and defense techniques in macOS. The purpose of this graph is to describe full relationships between malware/APT events, attack techniques, detection data artifacts, and analysis tools. Second, we develop a general remote administration tool (RAT) for red team emulation. The red team players can launch attacks, log attacks, and then map attacks to the MITRE ATT&CK matrix by using this tool. Third, we develop a toolkit for blue team evaluation by leveraging

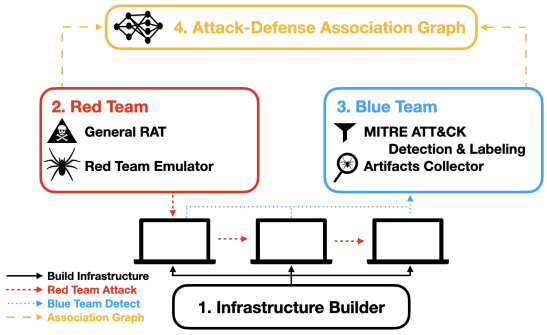

Figure 1: The architecture of our proposed cyber range.

open-source tools. The blue team players can collect artifacts, label artifacts with MITRE ATT&CK ID, and then evaluate their detection tools by using the toolkit. By combining the three components, we process red team logs and blue team reports, and then generate a comprehensive attack-defense association graph for users to easily identify the relationships between involved parties. With our proposed cyber range, it would be easier for security practitioners to evaluate the performance of red teams and blue teams.

2 ATTACK-DEFENSE ASSOCIATION GRAPH

The core of the association graph is the MITRE ATT&CK matrix. MITRE ATT&CK matrix is a public adversary technique database. Based on real-world observations in malware and APT reports, MITRE ATT&CK matrix systematic summarizes and enumerates adversary tactics and techniques. It's techniques cover most of the adversary techniques involved in the whole adversary life cycle. Since MITRE ATT&CK has become the de facto standard for developing threat models and methodologies in security community. We use it to detect and label attacks.

We build an attack-defense association graph based on MITRE ATT&CK matrix for evaluating red and blue teams. A sample graph is depicted in Figure 2. The objective of this graph is to depict the relationships between attack and defense techniques. For the attack side, we have to identify involved attack techniques based on the MITRE ATT&CK matrix. For the defense side, we have to find useful detection and forensic tools and sort out the artifacts supported by them. A link for bridging the attack side and the defense side is added between an attack technique and an artifact if the artifact contains evidences for revealing the attack technique. For instance, an artifact *file operation event* could be used to detect the technique *T1105 remote file copy*. Therefore, a link is added between the aforementioned artifact and technique. There are lots of artifacts that can

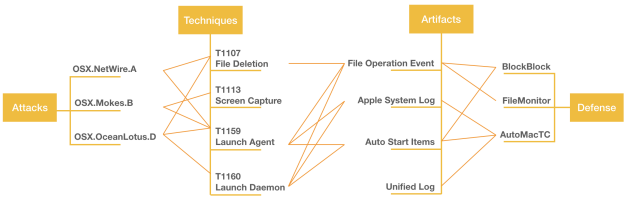

Figure 2: A sample attack-defense association graph.

be detected by forensic tools on macOS, including Apple system log, key-chain, unified Log, and so on. We can identify these artifacts and then track attacker activities.

There are several advantages to use our proposed association graph. From the perspective of red team, the attack side summarizes the techniques used by threat actors and malicious applications. Security practitioners can then identify commonly used techniques by observing the number of links connected from threat actors and malicious applications to their corresponding techniques. From the perspective of blue team, the relationships between detection and forensic tools and their supported artifacts show the capabilities of the tools. It provides an important information for security practitioners to decide how to select and deploy these tools.

3 RED TEAM

In this section, we first use a macOS malware sample to illustrate how to construct the attack side of an attack-defense association graph. We then develop our red team emulation tool based on techniques identified in the graph. The sample we chosen is OSX.NetWire.A discovered by Objective-See in 2019. It is a variant of OSX.NetWire, which is known as the first Trojan on macOS. The attacker used a phishing mail that contains a link to a malicious site. Upon clicking, the attacker sends the malware to the user's machine through a 0-day of Firefox (CVE-2019-11707). After exploited, it registers as LaunchAgent and Login Item to maintain its persistence. Finally, it provides several features for remote attackers such as shell login, screen capture, and keyboard event capture.

In the development of our red team emulation tool, we keep it modularized and compatible with the latest macOS. There are several challenges after a macOS updates. For instance, CVE used in the initial access stage may be fixed, and a malicious process may be forbidden to execute under a newer security mechanism. Thus, every single step in our storyline must be replaceable. We have to update it and look for the possible solution regularly. Our tool is mainly developed in Go language, and some plugins for privilege escalation and process injection are developed in C and Objective-C. The advantage of developing in Go language is that its binary is extremely complicated for analysis and there is a bunch of built-in packages for network communications. These two features make Go language better for developing malware.

The initial access of our tool is a CVE (CVE-2018-6574). It allows attacker to execute commands during gathering packages. We use it to download and execute the emulation tool. It is worth noting that it can bypass GateKeeper, a macOS security mechanism, because GateKeeper only sets the flag for files downloaded from normal means, not including command-line tools. After exploited, we register our tool as a user-level LaunchAgent. We do not register as a Login Item like OSX.NetWire.A did, since it may leave too many footprints for blue teams. Our tool connects to the C2 server through a socket. We provide an interactive shell on the server-side, and the red team can use it to send shell commands, take a screenshot, and perform specific attacks based on MITRE ATT&CK ID to victims. Our emulation tool also attempts to spread itself by scanning SSH configurations. Furthermore, it gains administrator privileges by spoofing privileged helper. After privilege escalation, it registers itself as a system-level LaunchDaemon and provide persistent service for red team.

4 BLUE TEAM

We survey several famous forensic tools for blue team, and summarize the artifacts supported by these tools. Then we integrate these forensic tools into our blue team toolkit, which could assist investigator to forensic. By the phase of investigation, our blue team toolkit has two phases: 1) Information Collection phase and 2) Malicious Activity Detection phase. The former phase collects the information either dynamic information during the attack occurs or forensic-based static information. The collected data then feed to the later phase, which contains several patterns that could identify possible malicious activities and label it with MITRE ATT&CK IDs.

During the Information Collection phase, our blue team toolkit composed of two classes of tools - static forensic tools and dynamic monitor tools. Static forensic tools can be further classify to two kinds. The first kind tools collect forensic evidence by gathering information from plists, SQLite databases and the local file system. The second kind of tools collect Apple's new logging system introduced since macOS 10.12. These tools complement each other and increase the visibility for our forensic.

The first kind of static tools include AutoMacTC [2], osquery [6], osxcollector [7]. AutoMacTC is easy to use and highly configurable, and it use modular framework to quickly add features and adapt changings on macOS. AutoMacTC collects wide range of macOS information from browser information such as downloads, history and browser profiles to system information such as lsof, netstat, pslist. For instance, AutoMacTC's autoruns module finds the application information (.plist) in LaunchAgents, LaunchDaemons and Startup Items. While these locations could be abused by adversary to achieve persistence. These information is highly valuable for forensic. Osquery is another tool in this type. While developed by Facebook since 2014, Osquery treats an OS as a relational database. Given a SQL-like query statement, osquery could retrieve system information. Thus osquery is highly customizable, interactive and possible to support different OSes. These tools help our blue team toolkit collect static information.

The second kind of static tools include Consolation 3, log & built-in Console.app, UnifiedLogReader. Consolation 3, log + Console.app (built-in) are essentially the same, Consolation 3 has a GUI front-end which helps user easily use various filter or switch and support other displaying styles. In the case that investigators want to programmable parse or search to the log, the log + Console.app is more suitable. On the other hand, Consolation 3 is easier to use in the case of manual analysis. The last one tool - UnifiedLogReader directly parse the unified log's database files. If the live system is

unavailable and only log file can be found, UnifiedLogReader could be a good choice. Thus our blue team toolkit includes the three tools for different use cases.

Complement the aforementioned static forensic tools, our blue team toolkit also integrates some dynamic monitor tools, such as build-in dtrace tools, kemon [3] ProcessMonitor and FileMonitor. The dtrace tool can snoop function calls to open and create, and it could also trace some I/O events. Kemon is the pre and post callback-based framework for macOS kernel monitoring. This system is a powerful framework to monitor the process and file events. Based on Apple's new Endpoint Security Framework, ProcessMonitor and FileMonitor provide basic but useful runtime information like pid, path, ancestor, arguments, code-signing and timestamp. However, dtrace and kemon require disabling SIP to perform their functionality. These tools are hard to deploy in real environments. Oppositely, ProcessMonitor and FileMonitor utilize build-in security framework and not need to disable SIP. But the framework only available after macOS 10.15, therefore cannot be deployed in old systems.

After preparing aforementioned tools, we can move to Malicious Activity Detection phase. Given the output from these tools, our framework provides some basic pattern-match rules to identify malicious activities. Identified malicious activities are labeled with ATT&CK labels. User can also define their customized rules. These predefined rules as well as user-customized rules could help investigator to complete their tasks.

5 EVALUATION

In this section, we use our red team emulation tool to construct a complete APT storyline and use our blue team toolkit to detect it as a showcase. At the beginning of the attack, the red team uses the exploitation of CVE-2018-6574 to build a package on Github. Upon getting our malicious package, the emulation tool is executed and copy itself to a hidden directory under the home directory of the victim. It also adds a plist file to register as a user-level LaunchAgent. Then, it connects to the C2 server through a socket, and the red team uses shell commands to gather information of the victim.

In the meanwhile, it scans SSH configuration files and copies itself to the remote victim with the SSH key recorded in the files. On the other side, the red team monitor the victim through process discovery and screenshot, and use privilege escalation plugin, such as an AppleScript, to pop up a spoofing privileged helper with Setting icon. Once the user authorizes, the system-level red team emulation tool is executed. It registers itself as a system-level LaunchDaemon immediately. The red team can then perform advanced operations as root through shell commands.

Table 1 presents the evaluation results. We list the techniques used in the storyline mentioned above and the detection result of our blue team toolkit. The columns "red team" and "blue team" summarize the support status of our attack and defense tools, respectively. The blue team result heavily depends on the completeness of available filter rules. Therefore, the unsupported part is because most existing detection rules mainly focus on the discovery stage. Although our framework records as many system information as possible in system log files, it is difficult to distinguish malicious artifacts from normal ones in the initial access stage and the privilege escalation stage. We leave the issue as one of our major future work.

Table 1: Evaluation results.

ATT&CK Techniques	Red Team	Blue Team
T1195 Supply Chain Compromise	O	X
T1155 AppleScript	O	X
T1059 Command-Line Interface	O	X
T1204 User Execution	O	X
T1064 Scripting	O	X
T1158 Hidden Files and Directories	O	X
T1159 Launch Agent	O	O
T1160 Launch Daemon	O	O
T1514 Elevated Execution with Prompt	O	X
T1144 Gatekeeper bypass	O	X
T1081 Credentials in Files	O	X
T1145 Private Keys	O	X
T1083 File and Directory Discovery	O	O
T1057 Process Discovery	O	O
T1033 System Owner/User Discovery	O	O
T1049 System Network Connections Discovery	O	O
T1069 Permission Groups Discovery	O	O
T1082 System Information Discovery	O	O
T1087 Account Discovery	O	O
T1135 Network Share Discovery	O	O
T1201 Password Policy Discovery	O	O
T1105 Remote File Copy	O	X
T1021 Remote Services	O	X
T1005 Data from Local System	O	X
T1113 Screen Capture	O	X
T1132 Data Encoding	O	X
T1071 Standard Application Layer Protocol	O	X
T1022 Data Encrypted	O	X
T1030 Data Transfer Size Limits	O	X
T1041 Exfiltration Over C2 Channel	O	X
T1485 Data Destruction	O	X
T1489 Service Stop	O	X
T1529 System Shutdown / Reboot	O	X

6 CONCLUSION

In order to improve both blue and red team's skill of macOS, we develop a cyber range system for macOS. At first, we survey and summarize many forensic tools to build an attack-defense association graph, this graph could be a guideline and assessment tool to evaluate performance of red/blue team. With red team emulation tool and blue team toolkit, the exercises can be conducted. In the end, we show how to utilize our cyber range to simulate an APT attack. Our cyber range system could be useful in red/blue team training, cyber exercises and security product testing.

REFERENCES

[1] Bitdefender. 2017. Dissecting the APT28 Mac OS X Payload. https://download.bitdefender.com/resources/files/News/CaseStudies/study/143/Bitdefender-Whitepaper-APT-Mac-A4-en-EN-web.pdf.

[2] CrowdStrike Holdings, Inc. 2019. AutoMacTC: Automated Mac Forensic Triage Collector. (2019). https://github.com/CrowdStrike/automactc.

[3] DiDi, Inc. 2014. An Open-Source Pre and Post Callback-Based Framework for macOS Kernel Monitoring. (2014). https://github.com/didi/kemon.

[4] Objective-See. 2019. Burned by Fire(fox). https://objective-see.com/blog/blog_0x43.html.

[5] Objective-See. 2019. Pass the AppleJeus. https://objective-see.com/blog/blog_0x49.html.

[6] osquery Project. 2014. Performant endpoint visibility. (2014). https://osquery.io/.

[7] Yelp, Inc. 2018. A forensic evidence collection & analysis toolkit for OS X. (2018). http://yelp.github.io/osxcollector.