Obfuscation, Stealth, and Non-Attribution in Automated Red Team Tools

Dwain Hembree, Alan Shaffer and Gurminder Singh
Naval Postgraduate School, Monterey, California, USA

dwain.k.hembree.mil@us.navy.mil
alan.shaffer@nps.edu
gsingh@nps.edu

Abstract: In the rapidly evolving landscape of cybersecurity, large military and government organizations face ever increasing persistent and sophisticated threats against their enterprise networks. The challenge of defending these networks is compounded by the increasing complexity and stealth of cyber-attacks, which can evade traditional security systems and measures, and remain undetected for extended periods. As a result, the need for advanced defensive strategies and tools that can keep pace with these evolving threats has never been more critical, however, current automated red teaming tools are limited in their ability to emulate advanced persistent threat (APT) behaviors. Supporting such behaviors in automated security assessments and tools can be helpful for improving organizations' cyber defense preparedness. This research demonstrates how obfuscation, stealth, and non-attribution techniques can be effectively automated into red teaming tools. We have enhanced our Cyber Automated Red Team Tool (CARTT) by integrating advanced evasion techniques to better simulate sophisticated cyber threats. By incorporating Metasploit Framework evasion modules and new custom Internet Control Message Protocol (ICMP) and Domain Name System (DNS) evasion capabilities into CARTT, its ability to evade detection by common security controls is significantly improved. In doing this, the research demonstrates how obfuscation, stealth, and non-attribution techniques can be effectively automated into red teaming tools. The enhanced CARTT has been tested in a virtualized operational environment, demonstrating its effectiveness in identifying vulnerabilities and assessing the robustness of security measures on a simulated enterprise network. The research results showed successful evasion of antivirus detection systems and covert data exfiltration using the newly implemented evasion techniques. The enhanced CARTT enables network managers as well as cybersecurity professionals to conduct more thorough evaluations of defense mechanisms against sophisticated threats, ultimately strengthening overall cybersecurity postures. The integration of sophisticated evasion techniques into CARTT represents a critical step in realizing the objectives of the DoD Cyber Strategy.

Keywords: Automated red team tools, Obfuscation, Stealth, Non-attribution, Evasion

1. Introduction

In the rapidly evolving landscape of cybersecurity, military and government networks face advanced persistent threats (APTs) and state-sponsored hackers. The increasing complexity of cyber-attacks demands advanced defensive strategies and tools to keep pace with these evolving threats. The United States Department of Defense (DoD) heavily relies on cyber red teams to improve and assess its cybersecurity posture, but these teams currently lack effective tools to realistically emulate APT behaviors (Department of Defense, 2021). According to the DoD Operational Test and Evaluation (DOT&E) Annual Report (2021), red teams face significant challenges when attempting to emulate advanced nation-state threats during exercises and operational tests. This lack of realism inhibits the ability of red teams to provide credible assessments of readiness against near-peer threats, highlighting the need for more sophisticated red teaming capabilities.

CARTT (Cyber Automated Red Team Tool) (Benito, 2022; Berrios 2020), developed at the Naval Postgraduate School in Monterey, CA, provides an automated platform for red teaming analysis, but its ability to evade detection and realistically portray APT-like threats is limited. This research enhances CARTT by integrating sophisticated evasion techniques.

The key contributions of this research include

- Integration of Metasploit Framework (MSF) evasion modules into CARTT to generate evasive payloads bypassing antivirus detection,
- Development of custom Internet Control Message Protocol (ICMP) and Domain Name System (DNS) evasion modules for covert data exfiltration, and
- Demonstration of CARTT's new capabilities through a simulated operational scenario.

This rest of the paper provides a background on obfuscation, stealth, and non-attribution techniques used by sophisticated threat actors, and examines existing automated red team tools. The design methodology explores the integration of advanced evasion techniques into CARTT, including MSF modules and custom ICMP

and DNS evasion modules. The implementation section demonstrates the new CARTT evasion capabilities in a simulated environment. Finally, we discuss conclusions and outline future work areas.

2. Background

2.1 Obfuscation

Obfuscation in cybersecurity refers to the deliberate act of making malicious code or payloads difficult to understand or interpret, with the goal of evading detection and analysis by security systems or researchers (Sharma et al., 2022). By using obfuscation, malware can bypass antivirus scanners, firewalls, and intrusion detection systems (IDS) that rely on recognizing known signatures (Kilic et al., 2019). This technique allows threat actors and APT groups to extend the lifespan of their tools and infrastructure by delaying detection and preventing timely remediation.

Common obfuscation techniques include encryption, encoding, packing, compression, polymorphism, and metamorphism (Nicho and Alkhateri, 2021; Samociuk, 2023; Sharma et al., 2022). Encryption and encoding transform a payload into ciphertext or alternative representations, making it harder to identify (Samociuk, 2023; Sharma et al., 2022). Packing and compression reduce the size of malware executables to obstruct static analysis (Nicho and Alkhateri, 2021). Polymorphic malware automatically generates new encrypted variants of itself, while metamorphic malware completely rewrites its structure with each iteration (Sharma et al., 2022).

Other methods such as instruction substitution and dead code insertion further complicate analysis efforts (Nicho and Alkhateri, 2021). These techniques create confusion, introduce seemingly valid but illogical code transitions, and cause disassemblers to make incorrect assumptions about the malware's execution flow. The diversity of obfuscation techniques enables malware authors to easily transform payloads, allowing malware to remain undetected for extended periods, be repurposed for future attacks, and overwhelm defender resources through exponential versioning.

2.2 Stealth

Stealth techniques are crucial for cyber threat actors, enabling them to operate undetected within compromised networks while achieving their objectives (Buchanan et al., 2020). The ability to operate covertly is essential as detection can lead to loss of access and prevention of further damage. Stealth enhances freedom of maneuver, helps avoid attribution, maximizes access duration, and conserves resources. Threat actors employ specific tactics to maintain stealth, including the use of legitimate credentials instead of deploying additional malware (Buchanan et al., 2020). Process injection, a technique used to insert malicious code into legitimate processes' memory space, is another common method (Sharma et al., 2022). Adversaries also abuse dual-use tools already present on victim systems, known as "living off the land," to blend in with legitimate administrative activity (Barr-Smith et al., 2021).

To counter analysis efforts, attackers employ anti-debugging, anti-virtualization, and anti-sandbox techniques (Afianian et al., 2018). These methods check for the presence of analysis environments and may alter behavior to evade detection (Afianian et al., 2018). Covert channels are used to maintain stealthy command and control (C2), often tunnelling traffic over legitimate protocols like DNS or HTTPS (Sharma et al., 2022). Techniques such as domain generation algorithms (DGAs) provide resilient and stealthy communication mechanisms, where attackers can dynamically generate domain names and constantly shift Internet Protocol (IP) addresses making it challenging for defenders to track and block malicious activities effectively (Sharma et al., 2022).

2.3 Non-Attribution

Non-attribution techniques are employed by threat actors to conceal their identity and obscure the origin of cyber operations, making it challenging to trace attacks back to the responsible party (Sharma et al., 2022). These techniques provide significant advantages, allowing threat actors to operate without fear of retaliation and repeatedly exploit targets. APT groups rely heavily on non-attribution to accomplish their objectives (Egloff and Smeets, 2023).

One common technique is packet spoofing, which involves altering source IP addresses and other packet header information to disguise the sender (Wheeler and Larsen, 2003). At the application layer, email spoofing enables phishing attacks by impersonating trusted senders (Wheeler and Larsen, 2003). Fast flux hosting and DGAs are employed during the initial access and C2 stages to enhance robustness and complicate tracking (Sharma et al., 2022). Threat actors also heavily rely on intermediary hosts and networks, acting as proxies or

stepping stones to effectively launder their traffic and make upstream attribution significantly more difficult (Wheeler and Larsen, 2003).

During C2, attackers leverage reputable web platforms and services, such as Microsoft Azure, to avoid detection (Sharma et al., 2022). When exfiltrating data, they use encrypted channels and covert communication tactics, including steganography and obscured channels like social media, image uploads, and DNS queries.

2.4 Automated Red Team Tools

Automated red team tools aim to streamline penetration testing, enabling efficient evaluation of network and system security. These tools allow under-resourced defenders to conduct self-evaluations and emulate complex attacks employed by state-sponsored groups. Several platforms have been developed to address this need. The Scalable Automated Vulnerability Scanning & Exploitation Tool (SAVE-T) builds upon the CALDERA framework, adding mechanisms for exploiting vulnerabilities using tools like Metasploit and Exploit-DB (Booz, 2020). SAVE-T implements exploit capabilities by fingerprinting services and sequentially querying the Metasploit API, demonstrating linear scalability on larger networks (Booz, 2020). Another tool, Automated Network Exploitation through Penetration Testing (ANEX), combines vulnerability assessment and penetration testing to automate network infiltration and path mapping (Dazet, 2016). ANEX utilizes open-source software like Metasploit, Armitage, Cortana, and Nmap to perform reconnaissance, exploitation, and reporting (Dazet, 2016).

CARTT provides red-teaming capability with minimal user expertise required (Benito, 2022). CARTT uses a client-server model comprising the Greenbone Vulnerability Management system, MSF, and a PHP server to enable automated vulnerability scanning, exploitation, and post-exploitation (Benito, 2022). This centralized configuration and automation of offensive tools like Metasploit enable even non-experts to conduct self-assessments of their networks' security posture.

Despite their advanced capabilities, these automated tools share common shortcomings in incorporating advanced evasion techniques. They often rely on well-known exploits and lack advanced obfuscation techniques, making them susceptible to detection by sophisticated defense systems. To enhance their effectiveness and realistically emulate threat behavior, integrating advanced obfuscation, stealth, and non-attribution techniques is crucial. For instance, CARTT currently relies on well-known Metasploit modules easily detected by security tools [2]. By incorporating Metasploit’s evasion modules to add obfuscation and stealth, CARTT can bypass common defensive systems to realistically simulate sophisticated attackers.

3. Design

3.1 CARTT Architecture

We expanded the CARTT architecture by incorporating MSF evasion modules to better simulate real-world cyber threats and assess an organization’s cybersecurity posture. These modules allow CARTT to circumvent target security controls and more successfully exploit vulnerabilities on targeted systems. To achieve this integration, we extended the CARTT architecture to interface with existing MSF evasion modules, as illustrated in Figure 1. The blocks with emboldened borders represent the new MSF evasion modules.

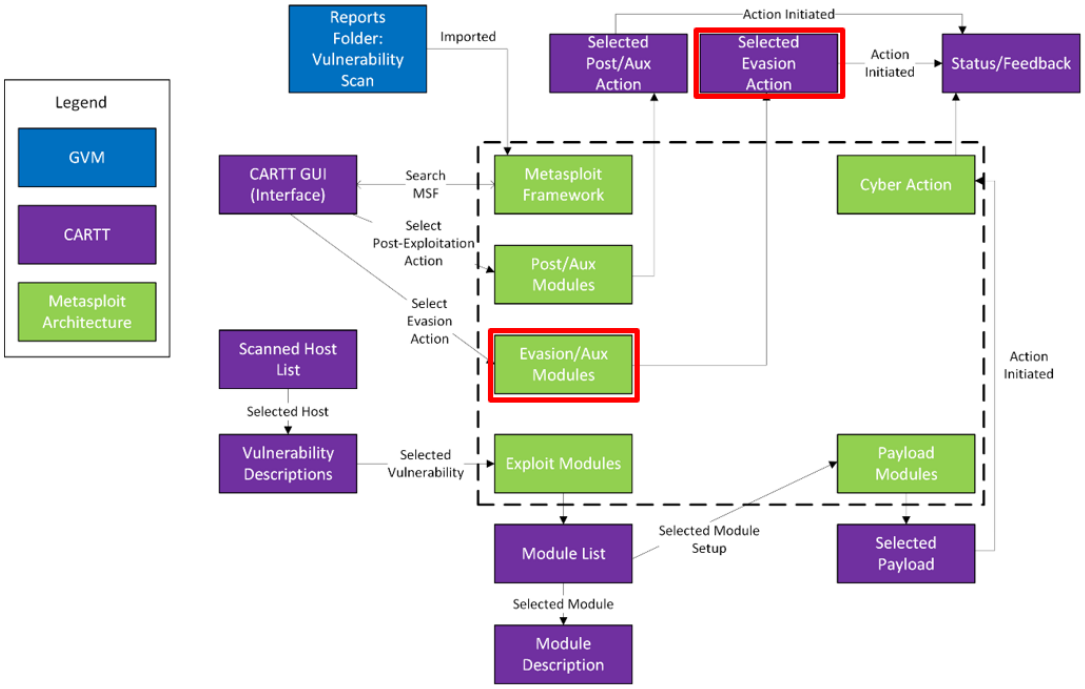

Figure 1: New CARTT Capability

In addition to integrating existing MSF evasion modules, the new CARTT design incorporates new custom ICMP and DNS evasion modules. These modules provide advanced capabilities for data exfiltration through unconventional channels. The ICMP evasion module allows CARTT to covertly transfer data by embedding it within ICMP packets, while the DNS evasion module enables data transfer through DNS queries and responses. The development of these custom modules required an in-depth understanding of network protocols and the ability to manipulate them for evasion purposes.

The process flow for evasion actions in CARTT is depicted in Figure 1. The CARTT Operator initiates an evasion action by selecting the desired option from the CARTT GUI. Based on the chosen evasion action, CARTT selects the appropriate MSF evasion or auxiliary module to execute the task. The evasion modules are specifically designed to bypass security mechanisms and ensure successful exploitation of vulnerabilities on the target host. Once the MSF evasion module has been selected, the CARTT Operator is prompted to confirm the evasion action, after which CARTT executes the selected module. The Operator receives real-time feedback on the results of the evasion action, allowing them to assess the effectiveness of the chosen technique.

3.2 MSF Evasion Modules

In 2018, Rapid7 developed Metasploit's first evasion modules to generate payloads that can bypass antivirus (AV) detection (Chen, 2018). These modules leverage a custom C compiler, random code generators, cryptographic functions, and anti-emulation functions. Four MSF evasion modules were selected for integration into CARTT: Evasion/windows/process_herpaderping, which uses "herpaderping" to obscure process behavior (Rapid7, n.d.); Evasion/windows/syscall_inject, which bypasses AV API hooking using direct system calls (Rapid7, n.d.); Evasion/windows/windows_defender_exe, which combines multiple techniques to bypass Windows Defender (Rapid7, n.d.); and Evasion/windows/windows_defender_js_hta, which exploits Windows Defender's JavaScript scanning weakness (InfosecMatter, n.d.).

We tested the selected evasion modules in a lab environment with Kali Linux (attacker) and Windows 10 (victim) virtual machines (VMs) to assess their effectiveness against Windows Defender with the results shown in Figure 2. All modules executed successfully with AV disabled. Against outdated AV (1/7/2021), process_herpaderping and syscall_inject evaded detection. However, no modules evaded current AV (3/27/2024). The VirusTotal scores, which represent the number of AV products that detected the payloads as malicious out of the total number of products tested, ranged from 31/60 to 50/72. These scores suggest that the evasion modules, in their current state, are not effective against a significant number of antivirus products and that improvements are needed to achieve broader evasion capabilities.

Evasion Module	Antivirus Disabled	Antivirus (out of date)	Antivirus (current)	VirusTotal
Process_herpaderping	Successful	Successful	Blocked	50/72
Syscall_inject	Successful	Successful	Blocked	31/72
Windows_defender_exe	Successful	Blocked	Blocked	48/73
Windows_defender_js_hta	Successful	Blocked	Blocked	31/60

Figure 2: Effectiveness of Metasploit Evasion Modules

3.3 ICMP Evasion Module

Data exfiltration is a key objective for many APT actors, who employ various techniques including covert channels and custom tools. This section examines a custom ICMP evasion module developed for CARTT, which provides a method for data exfiltration by sending data within ICMP packets. ICMP is a supporting protocol used by network devices to convey error messages and operational information (Postel, 1981). It includes several message types, such as Echo Request and Echo Reply. Each ICMP message has a header and a data section, with the data section's structure varying based on the message type (Postel, 1981).

The new evasion module operates with the existing "server/icmp_exfil" MSF auxiliary module, which acts as an ICMP exfiltration server (Riley, 2018). Key design considerations include compatibility with the MSF ICMP exfiltration server module, a simple interface for specifying target server IP, data to transfer, and payload size, splitting data into segments fitting within ICMP packet data field constraints, and encoding data to avoid flagging by security monitoring solutions. The module reads sensitive data, breaks it into smaller chunks, and sends each chunk as a separate ICMP packet to the server.

While functional, the module has limitations. ICMP is not a reliable transport protocol, potentially impacting data transfer. The module doesn't attempt to mimic legitimate ICMP exchanges, and detailed traffic inspection could detect embedded data payloads as anomalous. Monitoring for suspicious ICMP traffic and analyzing host logs for processes generating numerous ICMP messages can help mitigate this exfiltration technique.

3.4 DNS Evasion Module

DNS tunneling has emerged as a stealthy technique for data exfiltration in APT attacks, exploiting the trusted DNS protocol to bypass security controls (Nadler, Aminov and Shabtai, 2019). This section examines the custom DNS evasion module developed for CARTT, which establishes a communication channel between the CARTT server and compromised hosts. DNS is a hierarchical naming system that translates domain names into IP addresses, utilizing various record types such as A, AAAA, and TXT (Nadler, Aminov and Shabtai, 2019). Its pervasiveness in networks makes it an attractive channel for data exfiltration.

The CARTT DNS evasion module consists of client-side and server-side components. The client-side component, deployed on compromised hosts, encodes target data and constructs special DNS queries containing the encoded data in the subdomain portion. The server-side component, configured as an authoritative nameserver, extracts and decodes data from incoming DNS queries, storing it for later analysis. The module incorporates several stealth-enhancing techniques, including DGAs to evade domain blacklisting, data encoding to prevent leakage if intercepted, and adjustable timing intervals to mimic desired DNS traffic patterns.

Despite its effectiveness, DNS tunneling has limitations. It provides low bandwidth due to limited query size and encoding overhead, making it suitable primarily for small data exfiltration. Additionally, irregular traffic patterns generated by the module may be detectable by advanced monitoring solutions. In summary, the CARTT DNS evasion module demonstrates the potential of DNS as a channel for stealthy data exfiltration, leveraging the protocol's ability to bypass traditional security controls while highlighting the ongoing challenge of balancing effectiveness with detection avoidance in cyber operations.

4. Implementation

4.1 Workflow Extensions

The AV evasion file module extends CARTT's workflow by integrating selected MSF evasion modules into its existing architecture framework. This integration enables CARTT operators to generate evasive payloads directly within the tool, streamlining the process for creating antivirus-evading files for penetration testing and

security assessments. Figure 3 illustrates the process workflow for this module, where each block represents a distinct PHP page handling a specific step in the evasion file creation process. The modular structure ensures a clear delineation of tasks and a streamlined workflow from initial menu selection to results display.

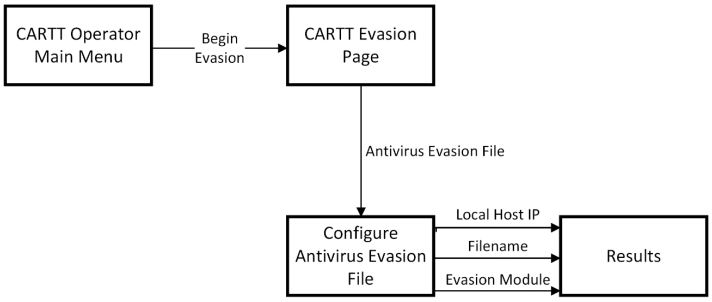

Figure 3: AV Evasion File Module Workflow

The process begins at the CARTT Operator Main Menu, where users select the "Begin Evasion". This action directs the user to the CARTT Evasion Page, where they choose "Create Antivirus Evasion File" to initiate the process. Users then proceed to configure the AV evasion file by inputting three essential parameters: the CARTT server's IP address (local host IP), a filename for the generated evasion file, and the specific MSF evasion module to be utilized. The chosen evasion module should align with identified vulnerabilities and target system characteristics from earlier analysis steps. To maintain system integrity and security, all user inputs undergo validation during this process.

Upon submission of the configuration, the system generates the evasion file and automatically redirects the user to a results page. This page clearly displays the success or failure status of the operation and, in the case of success, provides the storage location of the newly created evasion file. This process enables CARTT users to efficiently generate evasion payloads, enhancing the tool's overall effectiveness in penetration testing and security assessment scenarios.

The ICMP and DNS evasion clients represent extensions of CARTT's functionality, providing custom evasion capabilities that leverage ICMP and DNS for covert data exfiltration. Figure 4 illustrates the process workflow for these modules. The workflow begins at the CARTT Operator Main Menu, where the user selects the "Begin Evasion" option, leading to the CARTT Evasion Page. Here, they can choose either "Create ICMP Evasion Client" or "Create DNS Evasion Client." Users then input essential parameters including the CARTT server address (local host IP), a desired filename for the created file, the file type, and the specific file or directory to exfiltrate. Notably, entering 'DIR' as the file to exfiltrate causes the client to retrieve all files in the directory on the target where the file is stored.

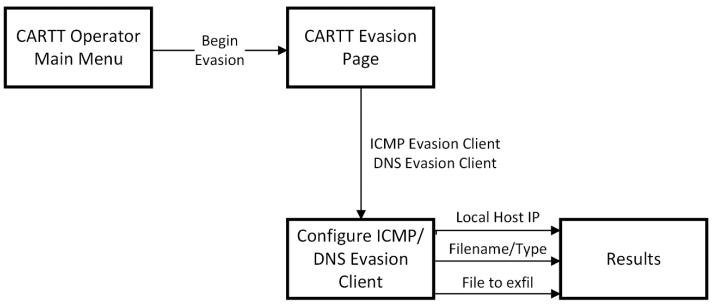

Figure 4: ICMP and DNS Evasion Client Workflow

After submission, the selected evasion client processes the inputs to generate a file for the designated exfiltration method. The file is generated using custom ICMP or DNS evasion modules discussed in sections 3.3 and 3.4, respectively. For ICMP evasion, the file incorporates functionality that encapsulates data within ICMP packets, while the DNS evasion file encodes data into DNS queries. The process concludes with a results page that displays the outcome of the file creation attempt and specifies where the newly created exfiltration file is stored on the local host.

CARTT's evasion features rely heavily on the listener and server components for efficient communication and data management. Figure 5 illustrates the workflow for these modules. The process begins at the CARTT

Operator Main Menu with the "Begin Evasion" option, leading to the CARTT Evasion Page which offers three listener/server options: "Start Antivirus Evasion Listener," "Start ICMP Evasion Server," and "Start DNS Evasion Server." Each option follows a similar high-level workflow of configuration, execution, and results reporting, but calls different MSF modules tailored to the specific evasion technique.

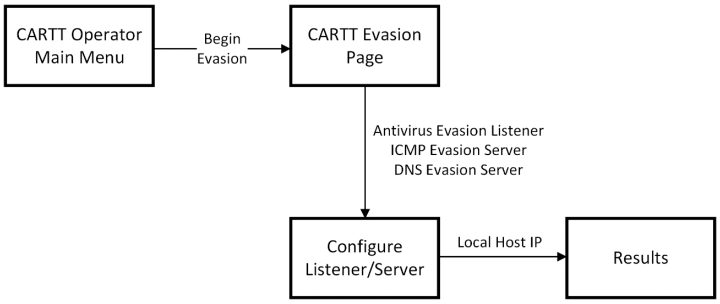

Figure 5: Listener and Server Modules Workflow

After selecting an option, users configure the chosen listener or server by inputting the local host IP (CARTT server's IP address). Once configuration is complete and inputs are validated, the chosen listener or server launches. For the AV evasion listener, the service begins listening for incoming connections from the evasion file. In the case of ICMP and DNS evasion servers, they start monitoring for specially crafted packets containing exfiltrated data. Upon terminating the service, users are presented with a results screen. For the AV evasion listener, this display confirms whether the evasion file successfully established a connection. With the ICMP and DNS evasion servers, the screen details the storage location of any exfiltrated files.

These modules are seamlessly integrated into CARTT's modular architecture, leveraging existing authentication and session management systems to ensure that only authorized operators can access their functionality. Throughout all processes, the system performs input validation to ensure all required fields are correctly entered, supporting the reliability and security of the operations. This comprehensive approach ensures that CARTT operators can effectively utilize the various evasion techniques in a structured and efficient manner, enhancing the tool's capabilities for penetration testing and security assessments.

4.2 CARTT Functionality

The CARTT evasion capabilities demonstration showcases the system's new features in a simulated operational environment. For this demonstration, Oracle VM VirtualBox was chosen as the virtualization platform, providing a flexible and isolated testing environment. The CARTT server, running on Ubuntu 20.04, was configured with a dual-network setup: an internal network for the target system and an internet-connected network for accessing resources. The target system, a Windows 10 virtual machine, represented a typical enterprise workstation.

The demonstration scenario revolves around a command that has recently addressed vulnerabilities identified in a red team assessment. To maintain a robust security posture, the cybersecurity team is tasked with conducting continuous assessments using CARTT. Their objectives include verifying AV effectiveness, evaluating firewall configurations, validating IDS performance, and assessing overall evasion detection capabilities.

The AV evasion scenario begins with the creation of an evasion file named "AV_test" using the Windows Defender JS HTA MSF evasion module. The left side of Figure 6 displays the results of this file creation process. The team then configures and starts the AV evasion listener on CARTT before transferring and executing the test file on the Windows 10 target. As shown in the right side of Figure 6, the AV evasion listener results indicate that the test file successfully evaded AV detection and established a connection to the target.

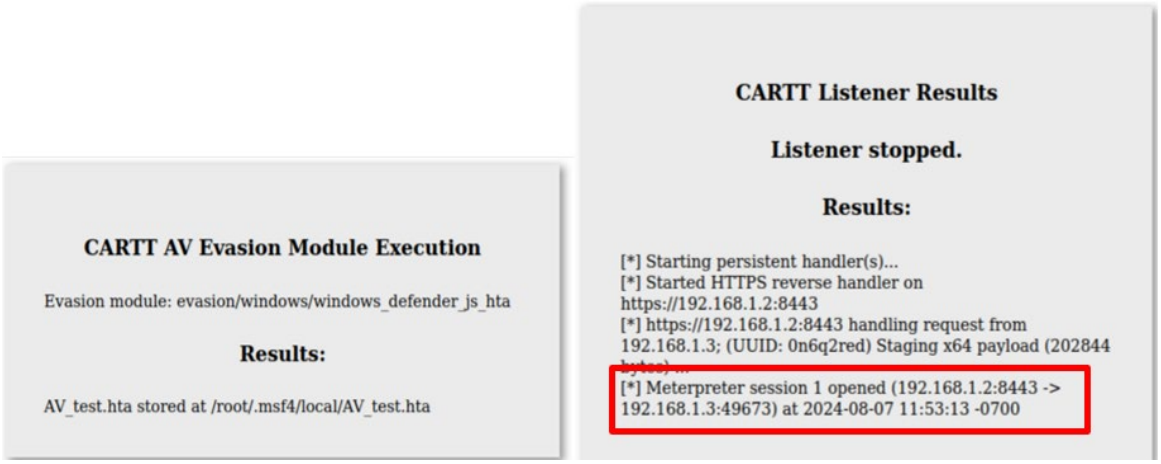

Figure 6: AV Evasion Module and Listener Results

Moving on to the ICMP evasion scenario, the team creates an ICMP evasion file named "ICMP_test" configured to exfiltrate a "test.txt" file from the target. The left side of Figure 7 illustrates the results of this ICMP evasion file creation. After configuring the ICMP evasion server on CARTT, the team deploys and executes the evasion file on the Windows 10 target. The success of the operation is confirmed by the CARTT ICMP server results, as shown in the right side of Figure 7, which indicate that the exfiltrated data was successfully received.

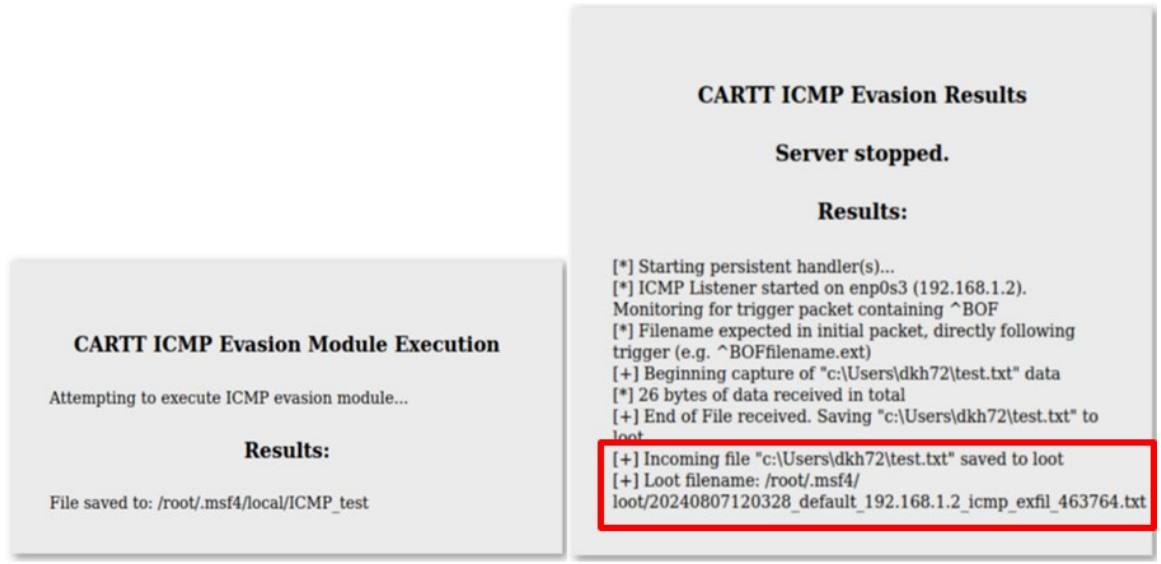

Figure 7: ICMP Evasion Client and Server Results

The final phase of the demonstration focuses on DNS evasion. The team generates a DNS evasion file named "DNS_test," also configured to exfiltrate the "test.txt" file. The left side of Figure 8 displays the results of this DNS evasion file creation. After establishing the DNS evasion server on CARTT, the team deploys and runs the evasion file on the test workstation. The success of the DNS evasion technique is verified by examining the CARTT DNS server results, as illustrated in the right side of Figure 8, which confirm that the targeted file was successfully exfiltrated and received on the local CARTT host.

Figure 8: DNS Evasion Client and Server Results

Throughout this comprehensive testing process, the cybersecurity team meticulously documents their findings, assessing the overall evasion detection capabilities of their security infrastructure. This systematic approach, enabled by CARTT's new evasion features, provides the command with valuable insights for continuously evaluating and enhancing their cybersecurity readiness. By simulating sophisticated attack scenarios that closely resemble the actions of real-world threat actors, the team can identify potential weaknesses in their defense-in-depth strategy and make necessary adjustments to bolster the command's overall security posture.

5. Conclusion

This research has expanded automated red teaming by integrating advanced evasion techniques into CARTT, significantly improving its ability to emulate sophisticated cyber threats and assess an organization's security posture. The integration of MSF evasion modules and custom developed ICMP and DNS evasion capabilities into CARTT have expanded its functionality, enabling it to more accurately simulate APT techniques. This provides cybersecurity professionals with a more comprehensive and realistic tool for conducting automated red team assessments. By incorporating obfuscation, stealth, and non-attribution techniques, CARTT can now better evade detection by common security controls such as antivirus software, firewalls, and intrusion detection/prevention systems.

The implementation and testing of the new CARTT capabilities in a simulated operational environment demonstrated their effectiveness in identifying potential vulnerabilities and assessing the robustness of security measures. The ability to generate evasive payloads, exfiltrate data using covert channels, and bypass signature-based detection systems provides valuable insights into an organization's security posture and highlights areas for improvement.

Future development of CARTT should prioritize its advanced stealth and non-attribution capabilities with a particular focus on post-operation clean-up. Implementing sophisticated clean-up routines would be crucial to maintaining adversary stealth and reducing the risk of detection during and after cyber assessments. This could include developing automated processes to clear logs on compromised systems, removing all files and artifacts associated with CARTT's operations, and eliminating any traces of the tool's activities on the target. These advancements would enable CARTT to conduct more realistic and thorough security assessments, particularly in scenarios where the need to avoid attribution and maintain long-term stealth is critical.

References

Afianian, A., Niksefat, S., Sadeghiyan, B. and Baptiste, D. (2018) "Malware dynamic analysis evasion techniques: A survey", [online], arXiv, http://arxiv.org/abs/1811.01190.

Barr-Smith, F., Ugarte-Pedrero, X., Graziano, M., Spolaor, R. and Martinovic, I. (2021) "Survivalism: Systematic analysis of windows malware living-off-the-land", [online], IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, https://ieeexplore.ieee.org/document/9519480/.

Bahrami, P.N., Dehghantanha, A., Dargahi, T., Parizi, R.M., Choo, K.K.R. and Javadi, H.H.S. (2019) "Cyber kill chain-based taxonomy of advanced persistent threat actors: Analogy of tactics, techniques, and procedures", [online], Journal of Information Processing Systems, https://www.researchgate.net/publication/350342843_Cyber_Kill_Chain-Based_Taxonomy_of_Advanced_Persistent_Threat_Actors_Analogy_of_Tactics_Techniques_and_Procedures.

Benito, R. (2022) "An automated post-exploitation model for offensive cyberspace operations", [online], M.S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, https://calhoun.nps.edu/server/api/core/bitstreams/ 17885f00-c82d-4cf7-a751-25a5a1304839/content.

Berrios, J.A. (2020) "A client/server model for automated red teaming", [online], M.S. thesis, Dept. of Comp. Sci., NPS, Monterey, CA, USA, https://calhoun.nps.edu/bitstream/handle/10945/66584/ 20Dec_Berrios_Joseph.pdf.

Booz, J. (2020) "Towards scalable automated vulnerability scanning & exploitation", [online], M.S. thesis, Dept. of Info. Sci., Carnegie Mellon University, Pittsburgh, PA, USA, https://kilthub.cmu.edu/articles/thesis/Towards_Scalable_Automated_Vulnerability_Scanning_Exploitation/12728360/1?file=24095777.

Buchanan, B., Bansemer, J., Cary, D., Lucas, J. and Musser, M. (2020) "Automating cyber attacks: Hype and reality", [online], Center for Security and Emerging Technology, https://cset.georgetown.edu/publication /automating-cyber-attacks/.

Chen, W. (2018) "Encapsulating antivirus (AV) evasion techniques in metasploit framework", [online], Rapid7, Technical Paper, https://www.rapid7.com/globalassets/_pdfs/whitepaperguide/rapid7-whitepaper-metasploit-framework-encapsulating-av-techniques.pdf.

Dazet, E. (2016) "ANEX: automated network exploitation through penetration testing", [online], M.S. thesis, Dept of Comp. Sci., California Polytechnic State University, San Luis Obispo, CA, USA, https://digitalcommons.calpoly.edu/theses/1592.

Department of Defense (2021) "Cyber assessment program", [online], Office of the Director, Operational Test and Evaluation, Washington, D.C., USA, Annual Report, https://www.dote.osd.mil/Portals/97/pub/reports/FY2021/other/2021cap.pdf.

Egloff, F.J. and Smeets, M. (2023) "Publicly attributing cyber attacks: A framework", [online], Journal of Strategic Studies, https://www.tandfonline.com/doi/full/10.1080/01402390.2021.1895117

InfosecMatter (n.d.) "Microsoft Windows defender evasive JS.Net and HTA - metasploit", [online], https://www.infosecmatter.com/metasploit-module-library/?mm=evasion/windows/.

Kilic, H., Katal, N.S. and Selcuk, A.A. (2019) "Evasion techniques efficiency over the IPS/IDS technology", [online], 4th International Conference on Computer Science and Engineering (UBMK), Samsun, Turkey, https://ieeexplore.ieee.org/document/8907177/.

Nadler, A., Aminov, A. and Shabtai, A. (2019) "Detection of malicious and low throughput data exfiltration over the DNS protocol", [online], Computers & Security, https://linkinghub.elsevier.com/retrieve/pii/ S0167404818304000.

Nicho, M. and Alkhateri, M. (2021) "Modeling evasive malware authoring techniques", [online], 5th Cyber Security in Networking Conference (CSNet), Abu Dhabi, United Arab Emirates, https://ieeexplore.ieee.org/ document/9614645/.

Postel, J. (1981) "Internet control message protocol", [online], RFC 792, http://www.ietf.org/rfc/rfc792.txt

Rapid7 (n.d.) "metasploit-framework/documentation/modules/evasion/windows/process_herpaderping.md", [online], GitHub, https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/evasion/windows/process_herpaderping.md.

Rapid7 (n.d.) "metasploit-framework/documentation/modules/evasion/windows/syscall_inject.md", [online], GitHub, https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/evasion/windows/syscall_inject.md.

Rapid7 (n.d.) "metasploit-framework/documentation/modules/evasion/windows/windows_defender_exe. md", [online], GitHub, https://github.com/rapid7/metasploit-framework/blob/master/documentation/modules/evasion/windows/windows_defender_exe.md.

Riley, C.J. (2018) "ICMP exfiltration service", [online], Rapid7 Vulnerability & Exploit Database, https://www.rapid7.com/db/modules/auxiliary/server/icmp_exfil/.

Samociuk, D. (2023) "Antivirus evasion methods in modern operating systems", [online], Applied Sciences, https://www.mdpi.com/2076-3417/13/8/5083.

Sharma, A., Gupta, B.B., Singh, A.K. and Saraswat, V.K. (2022) "Orchestration of APT malware evasive manoeuvers employed for eluding anti-virus and sandbox defense", [online], Computers & Security, https://linkinghub.elsevier.com/retrieve/pii/S0167404822000268.

Wheeler, D.A. and Larsen, G.N. (2003) "Techniques for cyber attack attribution", [online], Defense Technical Information Center, Fort Belvoir, VA, http://www.dtic.mil/docs/citations/ADA468859.