CS 6727 - CyberSecurity Practicum Final Report

Living off The Land Attacks: Applications IoC Generator

Submitted to:
Prof. Mustaque Ahamad

Author:
Vedika Bang, vedikabang@gatech.edu

Date: Dec 3, 2023

Index

Table of Contents

Table of Figures

Table of Tables

1. Abstract

The cybersecurity threat landscape is in a constant state of flux, posing ever-evolving challenges for security researchers and professionals in Digital Forensics and Incident Response (DFIR). One notable trend is the advancement of 'Living off the Land Attacks'. These attacks have become more refined and tailored to specific applications. To counter this, the project is developing an open-source tool tailored to the dynamic cyber threat environment. This tool will generate Indicators of Compromise (IoCs) and create detection rules by rigorously testing applications that are frequently exploited by threat actors for activities such as lateral movement, data exfiltration, and persistence.

In addition, the tool is developed to integrate seamlessly with Security Information and Event Management (SIEM) systems. This integration is key to enabling the creation of custom plugins, which allow security teams to automatically collect and analyze IoCs. This automation markedly reduces the time needed to detect and respond to threats in ongoing cybersecurity incidents. Moreover, this tool aids in the patching process during the initial stages of access, thereby significantly bolstering the overall security defenses.

2. Introduction

In this paper, we will refer to living off the land if only pre-installed (third-party RATs) software is used and no additional binary executables are installed or exploited onto the system by the attacker. RATs are generally called “Dual-use” tools, and Dual-use tools are tools that can be used by an attacker to perform actions that lead to their end goal. This technique is used by attackers to minimize the footprint and stay low-key in the system until the motive has been achieved. It has gained popularity over the years for the simplicity and no-investment principle it follows. Attackers using this approach use trusted off-the-shelf and preinstalled system tools to conduct their attacks. Many of these tools are used by system administration, and allowed by the EDR tools. Hence, even when logs are generated it can be difficult to spot the anomalies, and don’t really ring a bell which eventually extends the breakout time and attackers invade into the compromised infrastructure for lateral movement, data exfiltration and persistence.[2]

Initially, attackers exploited binaries, drivers, or scripts native to Microsoft Windows Operating Systems or Linux OS. These BASs are signed by trusted parties and easily evade the monitoring by EDRs or SIEMs. The list of LoLBAS[10]and GTFOBins[11] is never ending and always being updated by security researchers.[13] These Open Source projects have served well to many well-known SIEMs, EDRs for effective analysis. Furthermore, along with increased awareness, usage of anti-exploitation features such as data execution prevention (DEP), address space layout randomization (ASLR), control-flow integrity (CFI), and Anti-ROP has increased, and it has become harder for attackers to find new reliable vulnerabilities.[2] As it takes longer to find exploits, it makes them more expensive to use. Hence many attackers revert back to simple and proven methods such as spear-phishing emails and social engineering, where no exploits are needed.

In 2018, 40% of cybersecurity successful attacks were fileless malware and this number has increased to 71% in 2023. This alone explains deadlier and stealthy attacks are actually working out for attackers. Using RMM softwares as RATs isn't a new concept. Yet, it is still one of the most growing trends in cyber-security attacks! [3] For example, just a few months ago, ScreenConnect was used as a pawn for Hive Ransomware. [14] Documents with macros, VB scripts, PowerShell scripts, or the use of system commands, such as netsh commands, all fall under the living off the land specification. [21]

3. Literature Survey

3.1 Background

Remote Access Tools (RAT) came on the scene in the late 1990s or early aughts, and may have been first used as administrative tools—hence its other name, Remote Administrative Tool. But it quickly evolved backdoor capabilities and became stealthier and deadlier. BO2K, SubSeven, and Netbus are just some early examples of RATs. RATs are well understood and documented, and anti-virus software can spot the RAT's signature.[1] Typical Living off the Land Attack chain involves three stages -

1. **Incursion** - This could be achieved by exploiting a remote code execution (RCE) vulnerability to run shell code directly in memory. More commonly it is an email with a malicious script inside a document or hidden in another host file such as a LNK file. The threat may implement multiple stages with downloader or self-decrypting parts, each of which might follow living off the land techniques again. Another method is misusing system tools by simply logging in with a stolen or guessed password.[3]

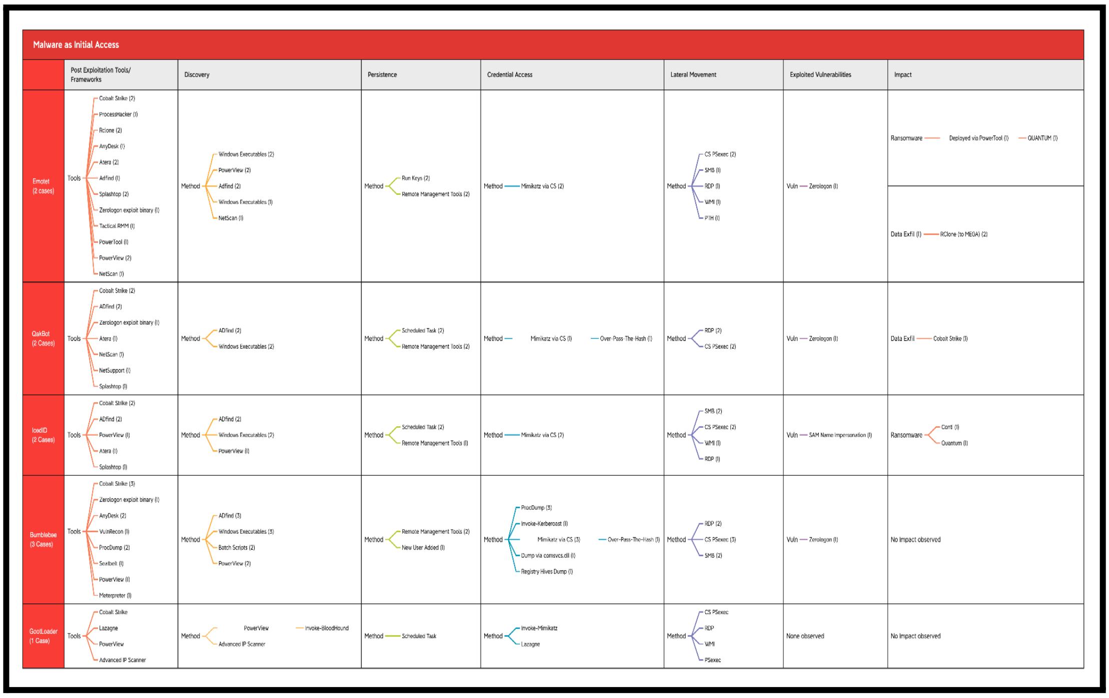
2. **Persistence** - Once the computer is compromised, stage two may or may not be fileless in regards to the persistence method. The threat may also not be persistent at all, depending on what the end goal is for the attacker.

3. **Payload** - The payload of the threat often makes use of dual-use tools.

3.2 Why use RATs?

Remote Access Tools (RATs) possess the capability to assume control of a system once access is granted. The transformation of these tools into Remote Access Trojans hinges on the identity and intent of the user in control. RATs enable a range of actions: they facilitate the uploading, downloading, or transferring of files across the same or different networks, depending on the specific functionalities of the application.[5] Furthermore, they allow the execution of PowerShell scripts, command execution, keystroke capture, keylogger installation, screenshot taking, video capturing, and comprehensive examination of file directories. These tools often represent the initial entry point for hackers into a target system, setting the stage for the deployment of 'Living off the Land' techniques.

One significant reason for the adoption of dual-use tools is their availability through free trials. Analyzing past cybersecurity incidents reveals a noteworthy trend: sometimes, fileless attacks are categorized as non-malware or malware-free. This classification arises in scenarios where only dual-use tools are employed, without the deployment of any malicious binary file. However, it's crucial to note that these attacks are not entirely fileless, as they involve the use of benign system tools. The distinctive feature of such attacks is their reliance not on custom-built malware binaries but rather on greyware tools or scripts. These can be termed asymptomatic attacks, as they often lack the conventional indicators of a cyber infection, such as the presence of a malicious file on the disk. This subtlety in their operation often makes them more challenging to detect and counteract, highlighting the need for advanced cybersecurity measures.



Malware as Initial Access

	Post Exploitation Tools/ Frameworks	Discovery	Persistence	Credential Access	Lateral Movement	Exploited Vulnerabilities	Impact
Emotet (2 cases)	Tools Cobalt Strike (2) ProcessHacker (1) Rclone (2) AnyDesk (1) Atera (2) Adfind (1) Splashtop (2) Zerologon exploit binary (1) Tactical RMM (1) PowerTool (1) PowerView (2) NetScan (1)	Method Windows Executables (2) PowerView (2) Adfind (2) Windows Executables (1) NetScan (1)	Method Run Keys (2) Remote Management Tools (2)	Method — Mimikatz via CS (2)	Method CS PSexec (2) SMB (1) RDP (1) WMI (1) PTH (1)	Vuln — Zerologon (1)	Ransomware — Deployed via PowerTool (1) — QUANTUM (1) Data Exfil (1) — RClone (to MEGA) (2)
QakBot (2 Cases)	Tools Cobalt Strike (2) ADfind (2) Zerologon exploit binary (1) Atera (1) NetScan (1) NetSupport (1) Splashtop (1)	Method ADfind (2) Windows Executables (2)	Method Scheduled Task (2) Remote Management Tools (2)	Method — Mimikatz via CS (1) — Over-Pass-The-Hash (1)	Method RDP (2) CS PSexec (2)	Vuln — Zerologon (1)	Data Exfil — Cobalt Strike (1)
IcedID (2 Cases)	Tools Cobalt Strike (2) ADfind (2) PowerView (1) Atera (1) Splashtop (1)	Method ADfind (2) Windows Executables (2) PowerView (1)	Method Scheduled Task (2) Remote Management Tools (1)	Method — Mimikatz via CS (2)	Method SMB (2) CS PSexec (2) WMI (1) RDP (1)	Vuln — SAM Name Impersonation (1)	Ransomware Conti (1) Quantum (1)
Bumblebee (3 Cases)	Tools Cobalt Strike (3) Zerologon exploit binary (1) AnyDesk (2) VulnRecon (1) ProcDump (2) Seatbelt (1) PowerView (1) Meterpreter (1)	Method ADfind (3) Windows Executables (3) Batch Scripts (2) PowerView (2)	Method Remote Management Tools (2) New User Added (1)	Method ProcDump (3) Invoke-Kerberoast (1) Mimikatz via CS (3) — Over-Pass-The-Hash (1) Dump via comsvcs.dll (1) Registry Hives Dump (1)	Method RDP (2) CS PSexec (3) SMB (2)	Vuln — Zerologon (1)	No Impact observed
GootLoader (1 Case)	Tools Cobalt Strike Lazagne PowerView Advanced IP Scanner	Method PowerView — Invoke-BloodHound Advanced IP Scanner	Method — Scheduled Task	Method Invoke-Mimikatz Lazagne	Method CS PSexec RDP WMI PSexec	None observed	No Impact observed

Figure 1: Usage of RMM tools in Post Exploitation Framework

3.3. Challenges in SIEM Detection

The 2023 Report on State of SIEM Detection Risk by CardinalOps offers a comprehensive analysis of SIEM (Security Information and Event Management) systems' effectiveness in current cybersecurity environments. SIEM systems only cover a limited range of MITRE ATT&CK techniques, with an average enterprise SIEM covering about 24% while having the potential to cover 94%. These systems often miss detections for a significant portion (76%) of adversary techniques due to challenges such as misconfigured data sources and parsing errors. This inefficiency is further compounded by the fact that many rules within SIEM systems are broken or ineffective. [22]

Based on extensive research and observations, there are five key challenges commonly encountered with SIEM systems. First, the effectiveness of a SIEM is heavily dependent on the quality of data it processes. Often, data sources are not entirely reliable, necessitating ongoing efforts in log collection, parsing, and filtering. Second, the deployment of a SIEM must be tailored to specific use cases; its functionality cannot be uniformly automated across varying scenarios, as each scenario presents unique and unpredictable characteristics. Third, having too much data can be counterproductive for SIEMs, causing delays in search results — a situation humorously dubbed as a "coffee break SIEM." Fourth, a significant challenge is the lack of sufficient context in the data provided by SIEMs, which complicates the task of analysts in making accurate assessments. Finally, SIEMs demand a considerable amount of maintenance, often diverting resources that could be more effectively allocated elsewhere in the organization.[6]

It's evident that while SIEM systems are a crucial component of modern cybersecurity strategies, they face significant challenges in effectively detecting sophisticated cyber threats, especially those employing RATs and dual-use tools. The complexity of SOCs, the uniqueness of each enterprise, and manual, error-prone processes further exacerbate these challenges. Therefore, It Is imperative to train SIEMs the correct way.

4. Concept, Methodology, Implementation, and Results

4.1 The Deceptive Nature of LoTL Attacks

Imagine a scenario – In a typical office, the IT team uses a tool called ScreenConnect for everyday tasks. But unknown to them, hackers have sneaked into their network. These hackers are cleverly using ScreenConnect, pretending to do regular IT work while secretly stealing important data. They're doing this so subtly that it blends in with normal network traffic, making it really hard to spot. Even though the company's security system, called a SIEM, is trying to find these hackers, it's not easy. The SIEM has to go through tons of data and the hackers are smart enough to hide their tracks. Owing to the SIEM’s challenges, Normally, the SIEM is good at its job, but in this case, it's struggling to tell the difference between the hackers using ScreenConnect and the regular IT work. And that’s a travesty! It prolongs the detection time, which tends to make attackers successful in their attacks.

The tool LOLAPP, inspired by LoLBAS and GTFOBins, is an open-source tool specifically designed to enhance the detection capabilities of existing SIEM and Endpoint Detection and Response (EDR) systems. By generating effective and precise Indicators of Compromise (IoCs) and tailored detection rules, this tool focuses on identifying the subtle signs of a Living off the Land (LotL) attack at the Application Level. It's developed to integrate seamlessly with any SIEM or EDR tool, augmenting their ability to sift through the noise and spot the discreet anomalies indicative of a sophisticated cyber attack.

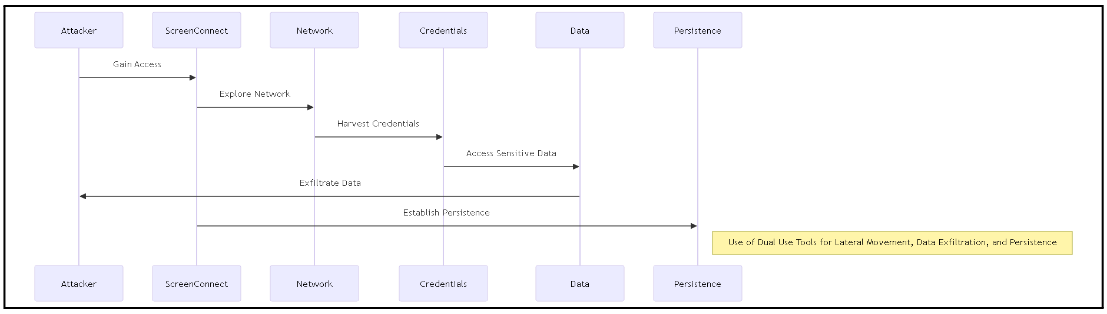


Figure 2: Use of Dual Use Tools for Living off the Land techniques

With this tool, organizations can significantly reduce the time it takes to detect and respond to stealthy infiltrations, especially those using dual-use tools like RMM software.

4.2 Methodology and Implementation

The project's development was strategically segmented into three critical phases to ensure precision and effectiveness in enhancing cybersecurity capabilities.

4.2.1 Phase 1: Testing

4.2.1.1 Criteria for Applications

1. Dual-use tools have a spectrum where it involves password-cracking tools, RMM tools, Network scanning tools, Vulnerability Scanner Tools,
2. For the PoC, I considered RMM tools. (See the future work). Applications were chosen based on their popularity amongst threat actors.
3. Over the span from July 2021 to June 2022, and the following year up to June 2023, the landscape of Remote Monitoring and Management (RMM) tools has seen notable shifts. AnyDesk has emerged as the predominant tool, with ConnectWise Control and Atera Agent also seeing substantial uptake. This trend underscores an increasing dependence on RMM tools within IT ecosystems.[3]
4. The period witnessed TeamViewer and Remote Desktop Plus maintaining their footholds, whereas solutions like RustDesk, Splashtop, FleetDeck, TightVNC, and N-able Remote Access Software rounded out the list of top RMM solutions. The rising utilization of these tools highlights their critical role in enabling efficient network management and remote support in a dynamic digital environment.[3]

4.2.1.2 Chosen Applications

1. **ScreenConnect:** ScreenConnect (now ConnectWise Control) offers robust remote control capabilities that, if compromised, can be exploited by attackers to discreetly infiltrate networks and manipulate systems without deploying additional malicious tools.[15]

2. **AnyDesk:** AnyDesk's lightweight and fast connection protocol can be abused by threat actors to gain remote access, enabling them to leverage the software's legitimate features for surveillance and data exfiltration in LOTA scenarios.[16]

3. **TeamViewer:** Often utilized for legitimate remote support, TeamViewer can be misused by attackers to gain remote access and execute commands, facilitating LOTA by blending in with normal administrative activities.[17]

4. **Atera:** As a comprehensive RMM (Remote Monitoring and Management) solution, Atera could be used by attackers to conduct LOTA by exploiting its monitoring and scripting tools to move laterally and maintain persistence in a target network.[18]

5. **NetSupport:** NetSupport Manager's extensive remote control and classroom management features, if exploited, can be repurposed for reconnaissance and lateral movement in LOTA, due to its deep access to system resources and network capabilities.[19]

4.2.1.3 Choosing Testing Tools and Environment

In the process of extracting artifacts and testing the environment, I utilized a collection of specialized tools, prioritizing those that do not compromise the integrity of the environment by leaving any residual data.

During the project, I encountered a bug in FTK Imager 4.7.1 that caused issues with exporting Windows log files. To circumvent this obstacle, I turned to Autopsy, which necessitated the use of dd images for analysis. Upon securing the raw images and logs, I leveraged a suite of tools specifically designed for meticulous artifact analysis.

AccessData Forensic Toolkit (FTK)	This is a comprehensive forensics solution which allows for the recovery and analysis of digital data. It is often used to find evidence in legal cases involving digital files.
Eric Zimmerman's Tools	A set of tools designed for Windows forensic analysis, including tools for parsing the registry, event logs, file system metadata, and more, aiding in thorough investigations.
Autopsy	An open-source platform that performs detailed analysis of various file systems and is widely used for digital forensics. It can recover photos, messages, and even call logs from devices.
MITRE ATT&CK	A knowledge base of adversary tactics and techniques based on real-world observations. It is used as a foundation for the development of specific threat models and methodologies in the cybersecurity community.
Registry Explorer	A tool that enhances the process of recovering and analyzing data from Windows Registry hives, providing more detail and context than traditional registry viewing methods.
Shellbags	A cross-platform, open-source shellbag parser that aids in forensic investigations by revealing user activities, such as folder opening, which can be pivotal in digital evidence.
Powershell and Python	For automating tasks, and parsing logs especially for NetSupport

Table 1: Suite of Testing Tools

Detection accuracy depends on the testing of the application across various versions of Windows, which also helps maintain the consistency and minimize the false positive rate. I used Windows 11, Windows 10 Vm and Windows 2019 Server based on their presence in the market as of Sept 5, 2023 - for uniformity of artifacts across the platform.[23]

4.2.1.4 Testing rounds

1. ***Same Network Testing:*** In this scenario, both the client and server components of the remote desktop applications are on the same network.
   For ScreenConnect, it was noted that the network artifacts, such as IP addresses, remained consistent regardless of network changes. This could suggest that ScreenConnect uses fixed endpoints or has a mechanism to maintain consistency in its network footprint. Hence, it can be monitored consistently.

2. ***Different Network Testing:*** Testing in different networks involves connecting the client and server from different network environments, such as different ISPs. I definitely expected variations in IoCs due to different routing, NAT behaviors, or firewalls that might alter the network traffic patterns.
   For ScreenConnect, the lack of change in network artifacts suggests robust connection methods, possibly using cloud infrastructure or dedicated relay servers that abstract away the underlying network differences.

3. ***VPN Testing:*** When using a VPN, the actual IP addresses are masked and traffic is routed through the VPN server. AnyDesk showed a change in IP addresses even within the same network when a VPN was used. This indicates that AnyDesk is sensitive to the actual network path taken, reflecting the new IP addresses assigned by the VPN. Such sensitivity to network changes can affect the predictability of IoCs, requiring more dynamic or behavior-based detection rules.

4.2.1.5 Analyzing artifacts

1. After extracting the artifacts for every round of testing, for every feature that application has to offer, analyzing artifacts was the most integral part.
2. Analyzing artifacts is like storytelling except this story could be live and it doesn't have to end. Windows artifacts tell a story about user behavior and system interactions within a Windows environment. Although adversaries try to destroy footprints, artifacts stay – window logs!
3. I divided artifacts into three categories based on the testing rounds and applications' behavior.
   a. ***Host-based artifact:***
   - Host-based artifacts include shellbags, windows event logs, Amcahe, Prefetch files, Registry, Windows search Index, etc.
   - These artifacts logged from operating systems, file timestamps, system databases, configuration changes, etc.

   b. ***Program-based artifact:***
   - These artifacts come from specific applications or programs installed on a device. They include application logs, history files, cache contents, or registry entries.
   - Program-based artifacts can reveal usage patterns, unauthorized access, or modifications to the software.
   - These artifacts uncover log Lateral movement, persistence in the system.

   c. ***Network-based artifact:***
   - These involve the capture and analysis of data as it travels across the network. Network logs, intrusion detection system alerts, and traffic patterns fall into this category.

- They are important for detecting suspicious activities like data exfiltration, unauthorized access, etc.
- These artifacts depending on the severity have potentials to become an IoC (Indicator of Compromise). Every IoC is important as it can help connect the dots, so IR can stop the breach/ransomware at the initial access.

4.2.2 Phase 2: Writing Detection Rules

The writing of effective detection rules in cybersecurity is a critical, data-driven process that necessitates a nuanced approach to balance accuracy, coverage, and performance. This section explains a structured methodology for developing these rules, focusing on understanding threats, defining detection criteria, and managing false positives and severity levels.

4.2.2.1 Sample Detection Rule generated from the CLI tool

```
title: AnyDesk Portable Executable Detected
status: experimental
description: Detects the AnyDesk portable executable in the user's AppData directory.
author: Vedika Bang
logsource:
   category: filesystem
   product: windows
detection:
   selection:
       FilePath:
           - 'C:\Users\*\AppData\Roaming\AnyDesk\'
           - 'C:\Users\*\Downloads\'
           - 'C:\Users\*\Desktop\'
   condition: selection
level: medium
```

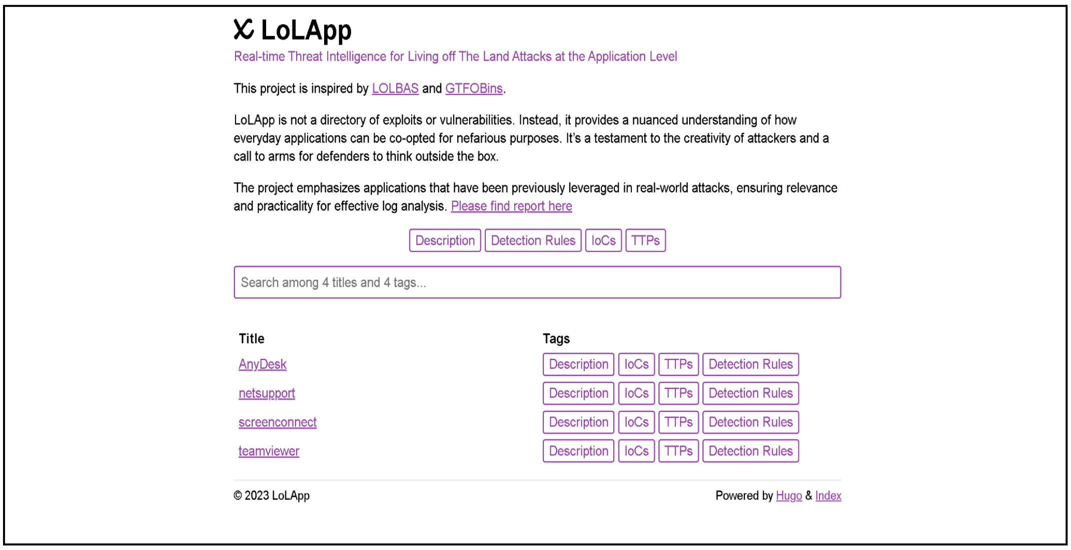
4.2.2.2 Methodology

1. ***Understanding the threat:***
   - *TTPs and IoCs*: Understanding the tactics, techniques, and procedures (TTPs) of adversaries by utilizing frameworks like MITRE ATT&CK to categorize and describe complex behaviors
   - Using analyzed IoCs such as Event IDs, IP addresses, or windows/program related artifacts
   - This avoids false negatives and improves accuracy.
2. ***Detection Criteria:*** This is the heart of the rule.
   - The selection clause identifies patterns to look for in the log data. - The condition stipulates what conditions must be met for an alert to trigger
3. False positive and Severity levels

4.2.3 Phase 3: Development of Tool

4.2.3.1 Development of WebUI backend

- The website [https://vedikabang.github.io/LoLApp/] is the backend for the Python-based command-line tool as it is easy to maintain a database online.
- The interface of the website is quite easy to use. I've created a template folder for adding new IoCs and Detection rules. This can be utilized by other security researchers to upload their findings. [https://github.com/VedikaBang/LoLApp]
- The website has 4 tags - Description, Detection Rules, IoCs and TTPs.



LoLApp

Real-time Threat Intelligence for Living off The Land Attacks at the Application Level

This project is inspired by LOLBAS and GTFOBins.

LoLApp is not a directory of exploits or vulnerabilities. Instead, it provides a nuanced understanding of how everyday applications can be co-opted for nefarious purposes. It's a testament to the creativity of attackers and a call to arms for defenders to think outside the box.

The project emphasizes applications that have been previously leveraged in real-world attacks, ensuring relevance and practicality for effective log analysis. Please find report here

Description | Detection Rules | IoCs | TTPs

Search among 4 titles and 4 tags...

Title	Tags
AnyDesk	Description IoCs TTPs Detection Rules
netsupport	Description IoCs TTPs Detection Rules
screenconnect	Description IoCs TTPs Detection Rules
teamviewer	Description IoCs TTPs Detection Rules

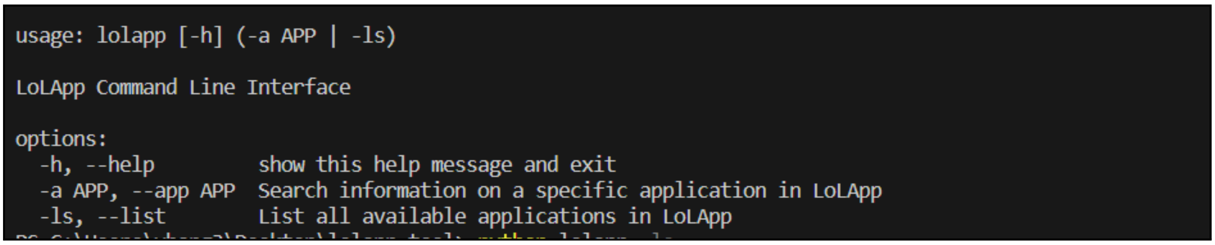
© 2023 LoLApp

Powered by Hugo & Index

Figure 3: Screenshot of the LoLApp website homepage

4.2.3.2 Python-based command-line tool

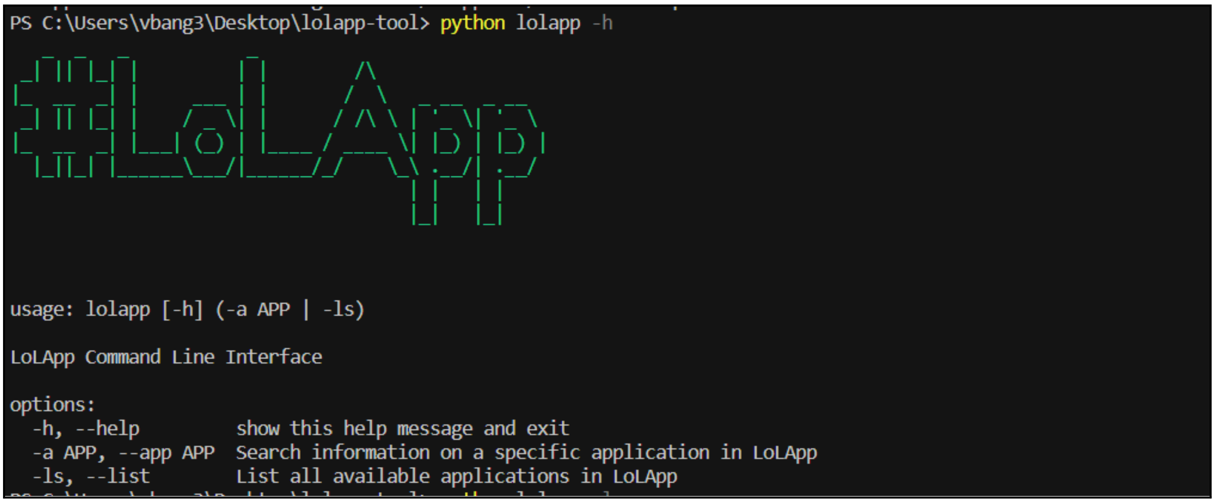
- The Python based command-line tool [https://github.com/VedikaBang/lolapp-tool] returns IoCs, and links to the Detection rules in the json format. Hence, iT can be directly integrated in any SIEM/EDR tool.
- Usage of the tool:
  - To get started: **`pip install -r requirements.txt`**
  - To run the tool: **`python lolapp -a <name of the tool>`**
  - To list all available applications: **`python lolapp -ls`**



```
usage: lolapp [-h] (-a APP | -ls)

LoLApp Command Line Interface

options:
  -h, --help          show this help message and exit
  -a APP, --app APP   Search information on a specific application in LoLApp
  -ls, --list         List all available applications in LoLApp
PS C:\Users\vbang3\Desktop\lolapp-tool> python lolapp -ls
```



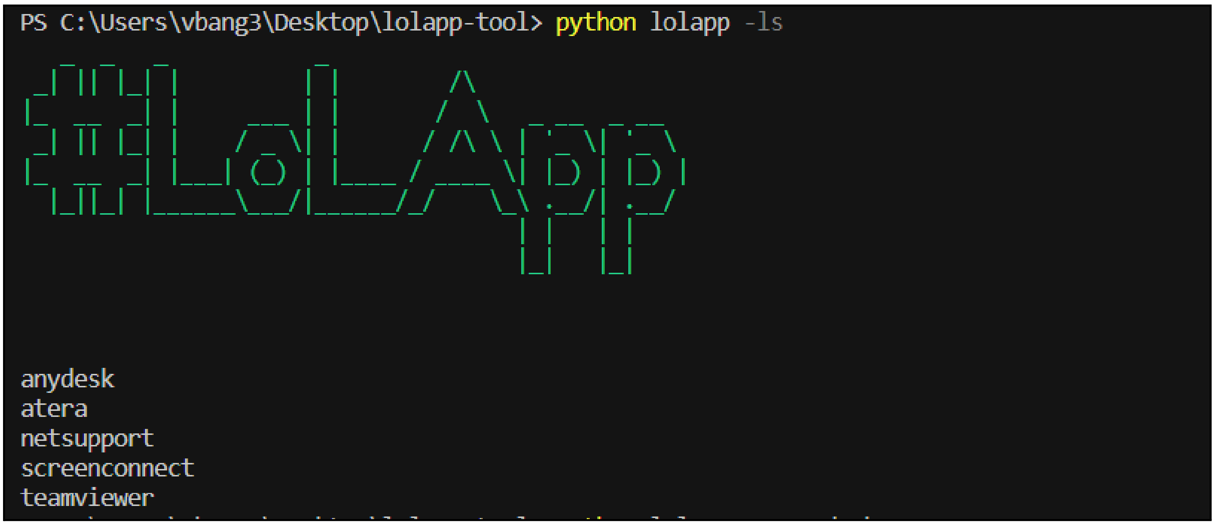
```
PS C:\Users\vbang3\Desktop\lolapp-tool> python lolapp -h
   _  _   _           _
 _| || |_| |         | |         /\
|_  __  _| |     ___ | |        /  \   _ __  _ __
 _| || |_| |    / _ \| |       / /\ \ | '_ \| '_ \
|_  __  _| |___| (_) | |____  / ____ \| |_) | |_) |
  |_||_| |______\___/|______|/_/    \_\ .__/| .__/
                                      | |   | |
                                      |_|   |_|


usage: lolapp [-h] (-a APP | -ls)

LoLApp Command Line Interface

options:
  -h, --help         show this help message and exit
  -a APP, --app APP  Search information on a specific application in LoLApp
  -ls, --list        List all available applications in LoLApp
```

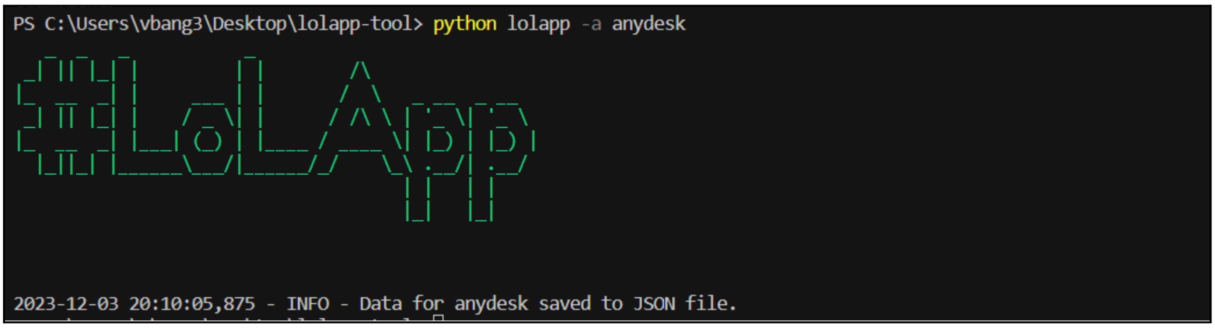
Figures 4 & 5: Usage menu and help menu of the CLI tool



```
PS C:\Users\vbang3\Desktop\lolapp-tool> python lolapp -ls
   _  _   _           _
 _| || |_| |         | |         /\
|_  __  _| |     ___ | |        /  \   _ __  _ __
 _| || |_| |    / _ \| |       / /\ \ | '_ \| '_ \
|_  __  _| |___| (_) | |____  / ____ \| |_) | |_) |
  |_||_| |______\___/|______|/_/    \_\ .__/| .__/
                                      | |   | |
                                      |_|   |_|


anydesk
atera
netsupport
screenconnect
teamviewer
```

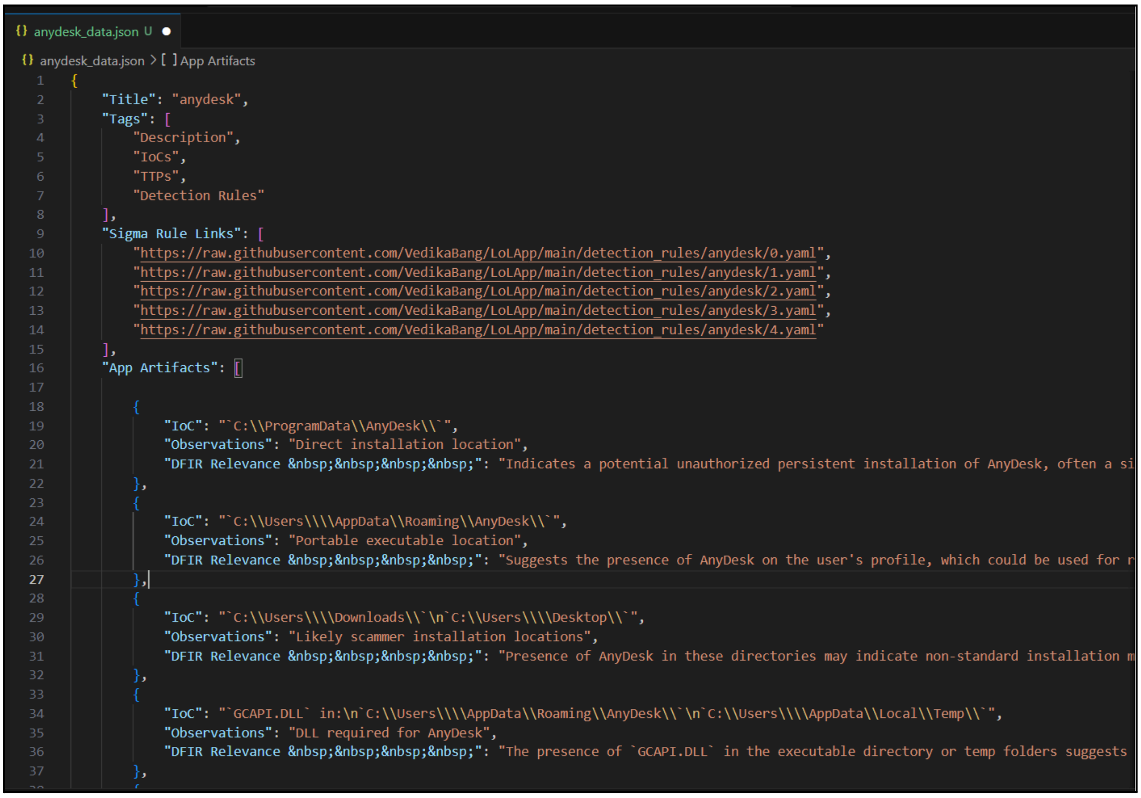
Figure 6: List of available applications



```
PS C:\Users\vbang3\Desktop\lolapp-tool> python lolapp -a anydesk
   _  _   _           _
 _| || |_| |         | |         /\
|_  __  _| |     ___ | |        /  \   _ __  _ __
 _| || |_| |    / _ \| |       / /\ \ | '_ \| '_ \
|_  __  _| |___| (_) | |____  / ____ \| |_) | |_) |
  |_||_| |______\___/|______|/_/    \_\ .__/| .__/
                                      | |   | |
                                      |_|   |_|


2023-12-03 20:10:05,875 - INFO - Data for anydesk saved to JSON file.
```

Figure 7: Tool output saved to a JSON file



```
{
    "Title": "anydesk",
    "Tags": [
        "Description",
        "IoCs",
        "TTPs",
        "Detection Rules"
    ],
    "Sigma Rule Links": [
        "https://raw.githubusercontent.com/VedikaBang/LoLApp/main/detection_rules/anydesk/0.yaml",
        "https://raw.githubusercontent.com/VedikaBang/LoLApp/main/detection_rules/anydesk/1.yaml",
        "https://raw.githubusercontent.com/VedikaBang/LoLApp/main/detection_rules/anydesk/2.yaml",
        "https://raw.githubusercontent.com/VedikaBang/LoLApp/main/detection_rules/anydesk/3.yaml",
        "https://raw.githubusercontent.com/VedikaBang/LoLApp/main/detection_rules/anydesk/4.yaml"
    ],
    "App Artifacts": [

        {
            "IoC": "`C:\\ProgramData\\AnyDesk\\`",
            "Observations": "Direct installation location",
            "DFIR Relevance &nbsp;&nbsp;&nbsp;&nbsp;": "Indicates a potential unauthorized persistent installation of AnyDesk, often a si
        },
        {
            "IoC": "`C:\\Users\\\\AppData\\Roaming\\AnyDesk\\`",
            "Observations": "Portable executable location",
            "DFIR Relevance &nbsp;&nbsp;&nbsp;&nbsp;": "Suggests the presence of AnyDesk on the user's profile, which could be used for r
        },
        {
            "IoC": "`C:\\Users\\\\Downloads\\`\n`C:\\Users\\\\Desktop\\`",
            "Observations": "Likely scammer installation locations",
            "DFIR Relevance &nbsp;&nbsp;&nbsp;&nbsp;": "Presence of AnyDesk in these directories may indicate non-standard installation m
        },
        {
            "IoC": "`GCAPI.DLL` in:\n`C:\\Users\\\\AppData\\Roaming\\AnyDesk\\`\n`C:\\Users\\\\AppData\\Local\\Temp\\`",
            "Observations": "DLL required for AnyDesk",
            "DFIR Relevance &nbsp;&nbsp;&nbsp;&nbsp;": "The presence of `GCAPI.DLL` in the executable directory or temp folders suggests
        },
```

Figure 8: JSON output generated by the tool for AnyDesk

5. Evaluation Matrix

Firstly, the generation of Indicators of Compromise (IoCs) is a key criterion. An effective tool should have the capability to generate potential IoCs that are robust and can be seamlessly integrated with any existing Endpoint Detection and Response (EDR) or Security Information and Event Management (SIEM) tools. LoLApp generates a JSON file including Detection Rules (Sigma Rules) and IoCs which can be seamlessly integrated with SIEM/EDR tools.

Next, the tool's capacity to create and implement detection rules. The flexibility and compatibility of these rules are crucial for a tool to be considered adaptable and useful in different security environments.

Another important aspect is the elevation of threat intelligence feeds. Integration with open-source projects is also a vital component. The tool's ability to push information to open-source projects like PyMISP[20] and IntelOwl[21] indicates its contribution to the broader cybersecurity community. 60% of pull requests have been accepted for PyMISP. Such integration facilitates collaboration, sharing of threat intelligence, and collective improvement of security measures.

Tool Name	**Description**	**Pros**	**False Positives**	**Ease of Use**
Surveyor [8]	Surveyor is a Python utility that queries Endpoint Detection and Response (EDR) products and summarizes the results.	Python-based, potentially more customizable and scriptable.	Depends on Carbon Black	May require Python knowledge and hands-on management.
LoLApp	An open-source tool aimed at addressing evolving cybersecurity threats by generating Indicators of Compromise (IoCs) and detection rules. It integrates with SIEM systems for creating custom plugins for analyzing IoCs.	Open-source, adaptable to evolving threats, SIEM integration for creating and analyzing IoCs.	Manual Testing - Minimum false positives. Integrated with Hive to get the best outcome.	Extremely easy to use and integrate.
Destra By IBM [7]	IBM Security ReaQta provides a unique feature called Detection Strategies (DeStra) that allows security operators to write custom detections, response rules and use cases to defend against advanced persistent threats (APTs), and to create highly-customized detection scenarios	Dynamic execution- They are also capable of identifying and responding to new defined behavior as-it-happens.	These detection rules are executed directly on the endpoint. A detection rule 'binds' to one or multiple events.	Currently, Destra cannot be tested offline before pushing them to the agents, therefore it is advised to test it in a dedicated environment. Difficult to use..

Table 2: Comparison with other existing solutions

6. Limitations & Challenges

The LoLApp currently has the following limitations:

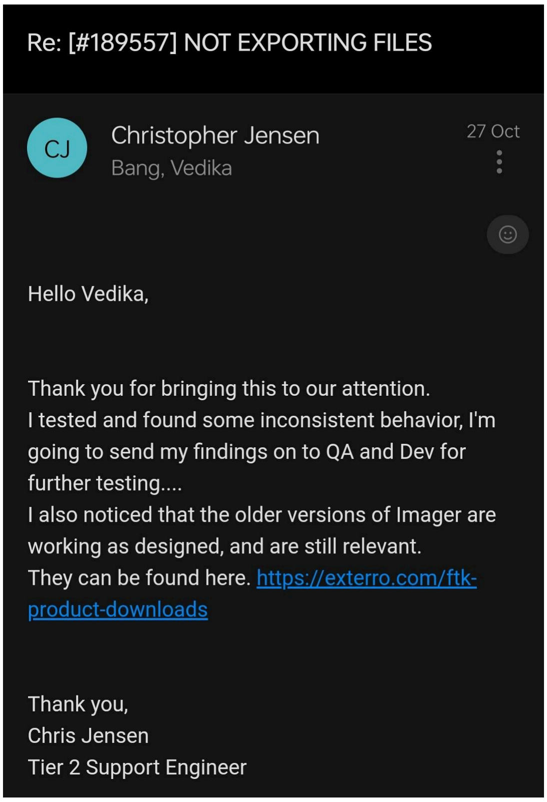
1. **Cross-Platform Support:** lolapp-tool currently exhibits optimized performance primarily on Windows-based systems. This presents a limitation in environments where Linux and MacOS are prevalent.

2. **OWASP Vulnerability Analysis:** While lolapp-tool generates IoC and detection rules based on artifacts, there is a limitation in its ability to specifically analyze applications for potential OWASP vulnerabilities. For instance, ScreenConnect was vulnerable to XSS attacks.[10]

3. **Program Analysis and IoC Extraction:** The current program analysis module is not as efficient in automating the extraction of IoCs while maintaining a minimum number of false positives. The precision of IoC extraction is paramount to avoid the costly distraction of chasing erroneous leads and to ensure that true threats are promptly and accurately identified.

4. **Comprehensive Security Tools Integration:** The toolset does not currently include password cracking utilities, network scanning tools, or vulnerability scanners. This represents a limitation in conducting a full spectrum security assessment, as these tools are fundamental to identifying and mitigating a wide range of cybersecurity threats.

While working on the PoC, I encountered myriad bugs in getting the testing environment. One of the more difficult to solve aspects, as mentioned before, was the bugs encountered with FTK imager. Here's a screenshot acknowledging the bug in the product:



Re: [#189557] NOT EXPORTING FILES

Christopher Jensen
Bang, Vedika

27 Oct

Hello Vedika,

Thank you for bringing this to our attention.
I tested and found some inconsistent behavior, I'm going to send my findings on to QA and Dev for further testing....
I also noticed that the older versions of Imager are working as designed, and are still relevant.
They can be found here. https://exterro.com/ftk-product-downloads

Thank you,
Chris Jensen
Tier 2 Support Engineer

Figure 9: Bug in FTK Imager

7. Future Work

To overcome the aforementioned limitations, the following enhancements are proposed for future development:

1. **Enhanced Cross-Platform Functionality:** Efforts will be directed towards making the tool universally compatible with Linux and MacOS in addition to Windows. This will broaden the tool's applicability and ensure consistent security management across all platforms.

2. **OWASP Top Ten Integration:** I plan to integrate a dedicated module for analyzing and mitigating OWASP Top Ten vulnerabilities. This will allow for a more targeted and effective approach to web application security.

3. **Advanced IoC Analysis Algorithms:** Development will focus on refining the algorithms responsible for IoC extraction. By leveraging machine learning techniques, I aim to reduce false positives and improve the accuracy of threat detection.

4. **Incorporation of Additional Security Measures:** To provide an all-in-one security solution, I plan to explore the incorporation of password cracking, network scanning, and vulnerability scanning capabilities. This will enable my tool to offer a more comprehensive security assessment and streamline the threat mitigation process.

8. Case Study - ScreenConnect

To demonstrate how exactly IoCs can come into the picture and make or break any ransomware case. Let us consider a scenario.

8.1 Background

A financial corporation noticed unusual traffic patterns in their network monitoring system. An Incident Response (IR) consultant named Alex was brought in to investigate. Alex is renowned for her ability to think like an attacker, which has helped her unravel complex cyber incidents in the past.

8.2 Initial Findings

1. **Suspicious Download:** Alex starts by looking into logs and discovers a URL pointing to an executable for ScreenConnect, a legitimate remote access tool. This executable was downloaded onto a machine belonging to the finance department, and it wasn't part of the standard IT deployment.
   IoC:
   `https://<username>.screenconnect.com/Bin/ScreenConnect.ClientSetup.exe?e=Access&y=Guest`
   Here, the username belongs to the individual whose account is registered with ScreenConnect.

2. **Administrator Event:** The Security Event log shows an EventCode=100 indicating an administrative account accessed the system shortly after the ScreenConnect installation. The administrator account named "IDKwho" was supposed to be dormant.
   IoC:
   `Cloud Account Administrator Connected EventCode=100`

3. **File Manipulation**: EventCode=201 is triggered, showing file creation and execution activities in ConnectWiseControl\Files. These files include batch scripts and a mysterious payload with a .bin extension. File transfers via ScreenConnect could point to data exfiltration or unauthorized file access.
   IoC: `C:\Users\<user>\Documents\ConnectWiseControl\Files EventCode=201`

4. **Temp Files**: Execution from this directory could indicate malicious use of legitimate software for unauthorized actions.
   IoC: `C:\Users\<user>\Documents\ConnectWiseControl\Temp`

8.3 Unraveling the Incident

The Trap: Alex hypothesizes that the attackers leveraged the ScreenConnect tool to masquerade their malicious activities under a veil of legitimate software. She suspects the finance team member was socially engineered into downloading the tool under the pretext of an IT update.

Privilege Escalation: By correlating timestamps, Alex notices the IDKwho account's activity occurred suspiciously close to the ScreenConnect event. This account, which should be inactive, appears to have been re-enabled and used to execute high-privilege operations across the network.

Decoding the Scripts: Alex carefully examines the PowerShell scripts and decodes the binary files using forensic tools. She discovers the scripts are designed to move laterally across the network using stolen credentials, and the binaries are custom-built malware designed for data exfiltration.

IoC:

```
powershell.exe-NoProfile-NonInteractive-ExecutionPolicy
Unrestricted-FileC:\WINDOWS\TEMP\ScreenConnect\23.4.5.8571\f5955c63-3955-4c
4a-ba98-672d4d6291eerun.ps1  EventID 4103
```

The Plot Thickens: Investigating the network logs, Alex finds out that the IDKadmin account has been accessing multiple servers within the network, especially those hosting sensitive financial data.

8.4 Additional DFIR cases involving RMM tools

1. Remote Access Software, Technique T1219 - Enterprise | MITRE ATT&CK®
2. From ScreenConnect to Hive Ransomware in 61 hours - The DFIR Report
3. MuddyWater APT Uses ScreenConnect to Spy on Middle East Governments - SOC Prime
4. Malware on Trial - Blackpoint Cyber
5. Are Bad Guys Swapping TeamViewer For AnyDesk to install Blackheart Ransomware?
6. DarkSide Ransomware Gang: An Overview

9. Acknowledgements

I extend my deepest gratitude to Prof. Mustaque Ahamad, whose guidance and expertise were instrumental in the completion of this project. His insights and support as the project guide were invaluable. I would also like to express my sincere thanks to Prof. Vijay Madisetty for his invaluable advice and assistance throughout this endeavor. His contributions significantly enriched my learning experience. Lastly, I am grateful to Prof. Anant Lummis for his encouragement and valuable suggestions, which were crucial in shaping the direction and execution of this project. Their collective wisdom and support have been a pillar of strength throughout the course of this project.

References

[1] Penetration Testing Explained, Part II: RATs!

[2] istr-living-off-the-land-and-fileless-attack-techniques-en.pdf

[3] crowdstrike-2023-threat-hunting-report.pdf

[4] Guide to Securing Remote Access Software

[5]https://learn.microsoft.com/en-us/previous-versions/tn-archive/dd632947(v=technet.10)?redirectedfrom=MSDN

[6] Why a SIEM Won't Solve All Your Problems: 5 Common SIEM Issues

[7]Dual-intent tools commonly used by hackers and how to defend against them

[8] Surveyor: an Open Source Tool for Carbon Black Response Environments

[9] GitHub - redcanaryco/surveyor: A cross-platform baselining, threat hunting, and attack surface analysis tool for security teams.

[10] Validating the Bishop Fox Findings in ConnectWise Control | by Kyle Hanslovan | Huntress

[11] https://lolbas-project.github.io/

[12] https://gtfobins.github.io/

[13]Living Off The Land: Threat Research February 2022 Release | Splunk

[14] From ScreenConnect to Hive Ransomware in 61 hours - The DFIR Report

[15] https://screenconnect.connectwise.com/

[16] https://anydesk.com/en

[17] https://www.teamviewer.com/en-us/download/windows/

[18] https://www.atera.com/

[19] https://www.netsupportmanager.com/

[20] https://github.com/MISP/PyMISP

[21] https://intelowlproject.github.io/

[22] https://cardinalops.com/

[23] Windows operating system market share by version 2017-2023 | Statista