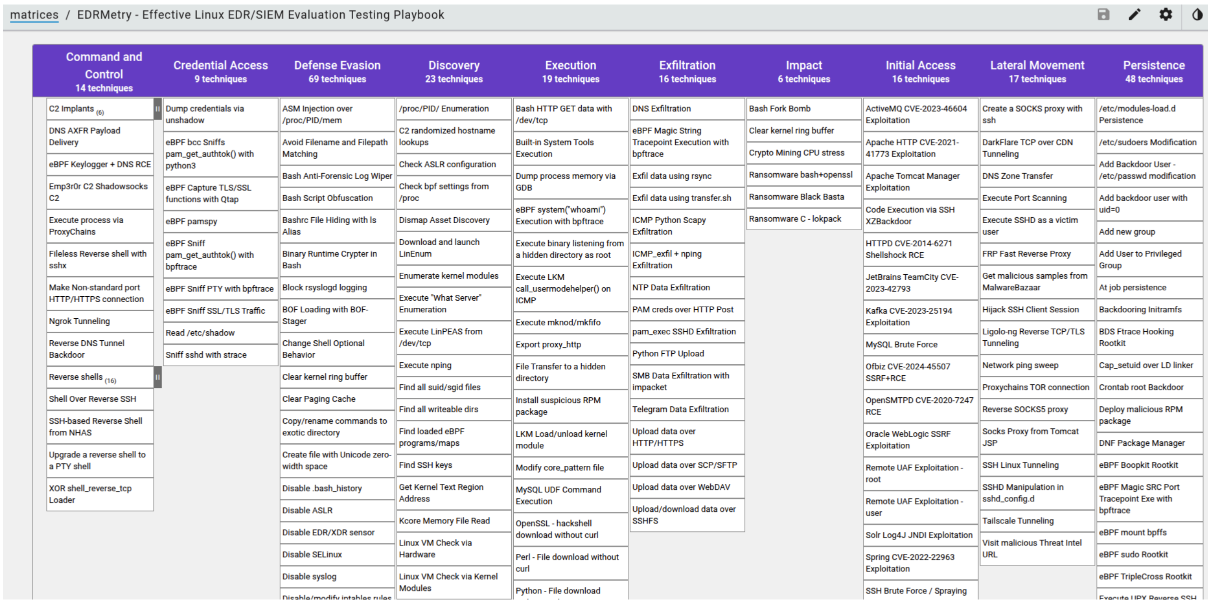
EDRmetry Linux Pulse - Automated Defense Validation through Adversary Emulation v1.2

Overview

EDRmetry Pulse is a user-friendly, automated tool for simulating adversary behavior on corporate Linux networks. It enables cybersecurity professionals to test Linux telemetry collection, evaluate detection capabilities, verify security controls, and enhance Linux incident response procedures. With a minimal learning curve and time-saving features, EDRmetry Pulse offers the quickest way to understand the true status of detection coverage and EDR/XDR/SIEM threat alerting in a continuous and automated format.

EDRmetry Pulse, built on the EDRmetry Matrix, automates the execution of over 300 offensive techniques (TTPs) derived from real-world Linux attack scenarios. It offers a comprehensive Linux offensive catalog, enabling users to launch individual “EDRmetries” or chain them together for advanced testing.



matrices / EDRMetry - Effective Linux EDR/SIEM Evaluation Testing Playbook

Command and Control 14 techniques	Credential Access 9 techniques	Defense Evasion 69 techniques	Discovery 23 techniques	Execution 19 techniques	Exfiltration 16 techniques	Impact 6 techniques	Initial Access 16 techniques	Lateral Movement 17 techniques	Persistence 48 techniques
C2 Implants (6)	Dump credentials via unshadow	ASM Injection over /proc/PID/mem	/proc/PID/ Enumeration	Bash HTTP GET data with /dev/tcp	DNS Exfiltration	Bash Fork Bomb	ActiveMQ CVE-2023-46604 Exploitation	Create a SOCKS proxy with ssh	/etc/modules-load.d Persistence
DNS AXFR Payload Delivery	eBPF bcc Sniffs pam_get_authtok() with python3	Avoid Filename and Filepath Matching	C2 randomized hostname lookups	Built-in System Tools Execution	eBPF Magic String Tracepoint Execution with bpftrace	Clear kernel ring buffer	Apache HTTP CVE-2021-41773 Exploitation	DarkFlare TCP over CDN Tunneling	/etc/sudoers Modification
eBPF Keylogger + DNS RCE	eBPF Capture TLS/SSL functions with Qtap	Bash Anti-Forensic Log Wiper	Check ASLR configuration	Dump process memory via GDB	Exfil data using rsync	Crypto Mining CPU stress	Apache Tomcat Manager Exploitation	DNS Zone Transfer	Add Backdoor User - /etc/passwd modification
Emp3r0r C2 Shadowsocks C2	eBPF pamspy	Bash Script Obfuscation	Check bpf settings from /proc	eBPF system("whoami") Execution with bpftrace	Exfil data using transfer.sh	Ransomware bash+openssl	Code Execution via SSH XZBackdoor	Execute Port Scanning	Add backdoor user with uid=0
Execute process via ProxyChains	eBPF Sniff pam_get_authtok() with bpftrace	Bashrc File Hiding with ls Alias	Dismap Asset Discovery	Execute binary listening from a hidden directory as root	ICMP Python Scapy Exfiltration	Ransomware Black Basta	HTTPD CVE-2014-6271 Shellshock RCE	Execute SSHD as a victim user	Add new group
Fileless Reverse shell with sshx	eBPF Sniff PTY with bpftrace	Binary Runtime Crypter in Bash	Download and launch LinEnum	Execute LKM call_usermodehelper() on ICMP	ICMP_exfil + nping Exfiltration	Ransomware C - lokpack	JetBrains TeamCity CVE-2023-42793	FRP Fast Reverse Proxy	Add User to Privileged Group
Make Non-standard port HTTP/HTTPS connection	eBPF Sniff SSL/TLS Traffic	Block rsyslogd logging	Enumerate kernel modules	Execute mknod/mkfifo	NTP Data Exfiltration		Kafka CVE-2023-25194 Exploitation	Get malicious samples from MalwareBazaar	At job persistence
Ngrok Tunneling	Read /etc/shadow	BOF Loading with BOF-Stager	Execute "What Server" Enumeration	Export proxy_http	PAM creds over HTTP Post		MySQL Brute Force	Hijack SSH Client Session	Backdooring Initramfs
Reverse DNS Tunnel Backdoor	Sniff sshd with strace	Change Shell Optional Behavior	Execute LinPEAS from /dev/tcp	File Transfer to a hidden directory	pam_exec SSHD Exfiltration		Ofbiz CVE-2024-45507 SSRF+RCE	Ligolo-ng Reverse TCP/TLS Tunneling	BDS Ftrace Hooking Rootkit
Reverse shells (16)		Clear kernel ring buffer	Execute nping	Install suspicious RPM package	Python FTP Upload		OpenSMTPD CVE-2020-7247 RCE	Network ping sweep	Cap_setuid over LD linker
Shell Over Reverse SSH		Clear Paging Cache	Find all suid/sgid files	LKM Load/unload kernel module	SMB Data Exfiltration with impacket		Oracle WebLogic SSRF Exploitation	Proxychains TOR connection	Crontab root Backdoor
SSH-based Reverse Shell from NHAS		Copy/rename commands to exotic directory	Find all writeable dirs	Modify core_pattern file	Telegram Data Exfiltration		Remote UAF Exploitation - root	Reverse SOCKS5 proxy	Deploy malicious RPM package
Upgrade a reverse shell to a PTY shell		Create file with Unicode zero-width space	Find loaded eBPF programs/maps	MySQL UDF Command Execution	Upload data over HTTP/HTTPS		Remote UAF Exploitation - user	Socks Proxy from Tomcat JSP	DNF Package Manager
XOR shell_reverse_tcp Loader		Disable .bash_history	Find SSH keys	OpenSSL - hackshell download without curl	Upload data over SCP/SFTP		Solr Log4J JNDI Exploitation	SSH Linux Tunneling	eBPF Boopkit Rootkit
		Disable ASLR	Get Kernel Text Region Address	Perl - File download without curl	Upload data over WebDAV		Spring CVE-2022-22963 Exploitation	SSHD Manipulation in sshd_config.d	eBPF Magic SRC Port Tracepoint Exe with bpftrace
		Disable EDR/XDR sensor	Kcore Memory File Read	Python - File download	Upload/download data over SSHFS		SSH Brute Force / Spraying	Tailscale Tunneling	eBPF mount bpffs
		Disable SELinux	Linux VM Check via Hardware					Visit malicious Threat Intel URL	eBPF sudo Rootkit
		Disable syslog	Linux VM Check via Kernel Modules						eBPF TripleCross Rootkit
		Disable/modify iptables rules							Execute UPX Reverse SSH

Key Values:

- **Automated Linux Offensive Testing, Smarter Linux Defensive Outcomes**
  Reduce manual effort by automating and chaining offensive techniques, allowing your team to validate the effectiveness of Linux EDR or Runtime Security engines in real-time. EDRmetry Pulse ensures your defenses aren't just theoretical—they're tested and proven.

- **See What Your SIEM Can't**
  Uncover blind spots in your detection pipelines, telemetry flows, and data source correlations. By replicating adversary behavior, EDRmetry Pulse highlights exactly where your current tools are falling short, so you can close the gap before attackers exploit it.

- **Elevate Your Incident Response**
  Use EDRmetry Pulse as a foundation for meaningful internal Red vs Blue team exercises. Improve coordination, sharpen your response strategy, and gain deeper insight into Linux-specific TTPs and forensic artifacts.

- **Enhance Detection Engineering and Threat Hunting**
  Focus your resources where they matter most—building better detections and expanding your threat-hunting capabilities. With EDRmetry Pulse, you maintain an active defense posture while aligning your efforts to real adversary behavior.

- **Make Informed Choices About Your EDR Stack**
  EDRmetry Pulse helps you define meaningful criteria for evaluating Linux EDR and SIEM solutions. Ask vendors the right questions, backed by technical insight and offense-driven evidence.

- **Cut the costs of periodic, external Red Team testing**
  Reduce the need for buying and using multiple offensive tools and expensive external services for Linux-oriented security testing by ongoing emulation of techniques that matter.

- **Save your time**
  All attack techniques are based on continuous active research for new offensive projects, attack techniques, research reports, including the analysis of CTI/APT reports and technical community write-ups, saving you the effort.

- **Test or Learn**
  Evaluate the EDR/SIEM effectiveness and visibility, or use EDRmetry for Red vs Blue team skill development in the next-generation, hands-on format of internal Linux security workshops/training

- **Linux Focus Only**
  The only adversary emulation platform dedicated to the Linux environment, providing

advanced techniques and generating low-level offensive events characterizing real malware

---

What makes EDRmetry different from others?

- **Full Range of Advanced Techniques** → Close the gaps in your Linux Security posture by emulating Linux offensive techniques faster and easier than ever before, allowing even less experienced users to understand advanced Linux security concepts

- **Ongoing Threat Intelligence Research** → A centralized, continuously updated knowledge base on the Linux threat ecosystem in auto-executable format, with Linux TTPs mapped to the MITRE Attack Framework.

- **Offensive TTPs as Code** → Full insight into the security tests source code, defined commands and snippets of code, with the possibility of easy customization

- **Code verification** → All source codes and open source projects used within the product have been verified for potential malware infections, allowing for a safe execution in your environment

- **Session Operations** → Engage with established C2 or reverse shell sessions, extend the execution contexts, observe the real-time output and execution status in an authentic shell environment via EDR-ID session tracking.

- **Agentless** → Support for on-premise and cloud Linux environments by easy deployment and fast integration via SSH communication channel

- **Recognition** → The proposed method and offensive content have been consistently evaluated as highly valuable during professional services and training sessions at prestigious cybersecurity conferences, such as Black Hat USA/Singapore, X33FCON, HITB, and also during private training for the biggest companies all over the world.

Technical Overview

All offensive test definitions, called "EDRmetries," are written in Ansible YAML for clear test logic, easy customization, and chaining. The Ansible engine manages execution via SSH communication, eliminating the need to install a dedicated agent. Ansible playbooks, stored in a central repository or local directory, define the core testing logic and are integrated into EDRmetry Pulse, a web application that enables on-demand test execution.

Current scope of EDRmetry tests

- True exploitation tests of included vulnerable network services and security misconfigurations to simulate attack behavior contextually

- User and Kernel-space tampering to evade traditional detection mechanisms
- eBPF Rootkits: Advanced syscall hooking for stealthy kernel and user space process memory manipulations, ex. code invocation on magic packet, file/process hiding, etc.

- Fileless Execution Techniques: Pure memory-based payloads, avoiding disk traces entirely, including binary and LKM loading from in-memory

- Generation of malicious network traffic using one-liners and full C2 frameworks like Sliver, Metasploit, Merlin, Mythic, and others

- Credential Access: steal user credentials by executing credential dumping techniques or password spray/brute force attacks

- Exfiltration over different network communication channels, including HTTPS, MTLS, DNS, ICMP, SSH, Websockets, NTP, FTP, and others

- Ready-to-use webshell implementations for various scenarios
- Living Off the Land Binaries from real-world attack scenarios
- Code Injection Techniques for breaking the parent-child process context.
- Encrypted Loaders: Testing encrypted payload execution for advanced evasion
- Ransomware Emulation: test simulations written in C, Python, and Bash, mimicking ransomware behaviors for validation under stress scenarios

---

Parametrization

EDRmetries support parameterization, allowing for easy adaptation to the client's environment. The basic design assumption was to minimize the number of static values, such as TARGET_INTERNAL_IP/TARGET_EXTERNAL_IP, C2_EXTERNAL_IP/C2_INTERNAL_IP, C2_SLIVER_MTLS_PORT, and others, such as commands executed from the RCE exploit level like PHP_SYSTEM_COMMANDS, or pid values for hiding, ex, MOUNT_PID_TO_HIDE, to name just a few.

Your testing environment

A dedicated testing environment, ideally located in a customer's dedicated VLAN, is based on the assumption that testing virtual machines are based on "Golden" images to achieve maximum compatibility with production systems. As part of the Preparation phase, EDRmetry provides a set of early-stage deployment automations that enable the local delivery of network services within vulnerable container images, binaries that facilitate LPE, and other types of security misconfigurations. The goal is to achieve a true contextual execution as close as possible to that observed in real-world attacks. Only in this way will you see whether your detection, telemetry, or alerting can actually work in the event of a real attack.

EDRmetry Hosts Inventory

- **EDRmetry Pulse VM:**
    - The Linux VM with an Ansible execution engine that includes a security tests repository
    - User-friendly web interface for running and managing tests
    - EDRmetry Matrix included as a public web service over https

- **TARGET_X VM (RHEL7/8/9)**
    - The main Linux VM under which attack emulations are carried out
    - Provides vulnerable services and security misconfigurations
    - This is the VM where you install your EDR/Runtime/SIEM agent
    - You can easily add many instances of TARGET_X, ex. RHEL7, RHEL8, RHEL9

- **DEVEL_X VM (RHEL7/8/9):**
    - A development DEVEL_X VM mirrors TARGET_X and is dedicated to the compilation of the included tests' source codes.
    - The idea is to provide compiled binaries, shared libraries, or LKM objects directly to the TARGET_X, avoiding local compilation.

- **C2_EXTERNAL VM (Kali Linux):**
    - external attacker machine dedicated to host payloads, handling egress reverse shell connections, installing C2 frameworks, pivoting over the public Internet, and many more

- **C2_INTERNAL VM (Kali Linux):**
    - An internal attacker machine dedicated to host payloads, handling local network reverse shell connections, installing C2 frameworks, pivoting over LAN/DMZ, and many more.

Core Features

- **Single and chained execution:**
  - Run a single test one by one, learn how it works and what detection artifacts it leaves behind, or build a custom, full attack scenario by combining and running multiple tests at once

- **Customizable views:**
  - All security tests are grouped by tactic, helping you navigate more easily

- **Parametrization:**
  - Every security test definition is based on global variables that you can easily adapt to your needs

- **Manual Interaction with sessions:**
  - Thanks to the session support of individual tests, it is possible to expand the execution context of the performed steps manually

- **Scheduled tests:**
  - Run ongoing tests at specific time intervals, as it is key to staying ahead of threats. Scheduled execution also allows for finding differences in the operational behavior of a given version of the EDR/SIEM engine, e.g., after an update

- **Teams and user roles:**
  - The EDRmetry Pulse web interface allows for creating dedicated user groups in the form of teams, taking into account an assigned set of permissions such as guest, task runner, manager, and admin

- **Reporting and statutes:**
  - Track the execution status, date, and history of executed tests in detail

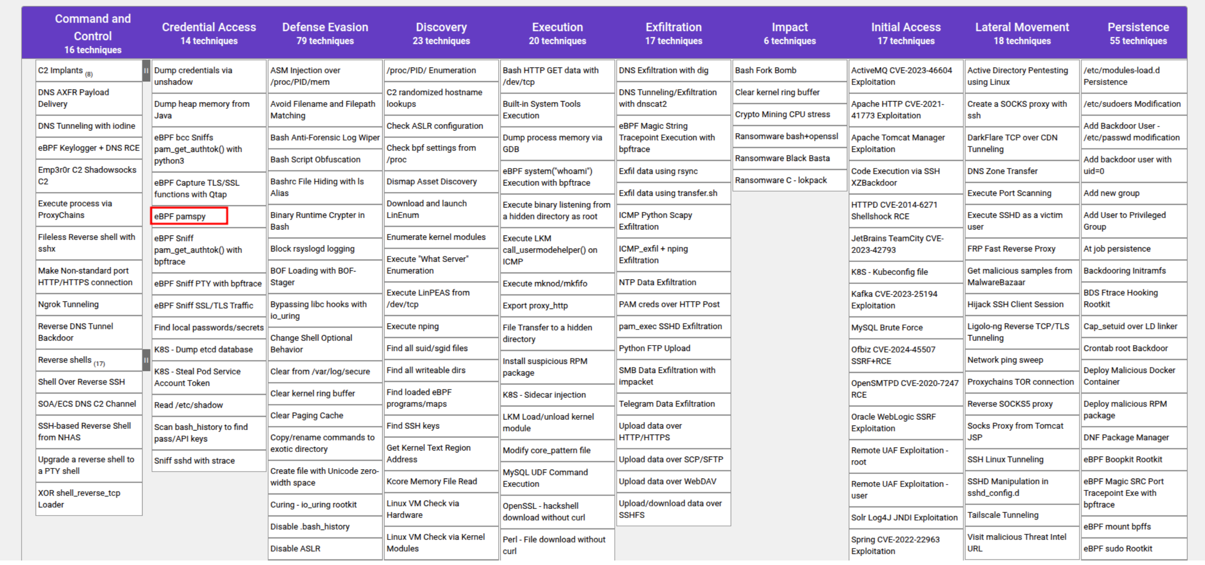
- **Updates:**
  - New EDRmetry test definitions delivered on an ongoing basis as part of the service through a dedicated git repository (by default, every 30 days)

An example flow

1. Choose and install on the TARGET VM the Linux EDR/Runtime Security/SIEM engine you want to evaluate.

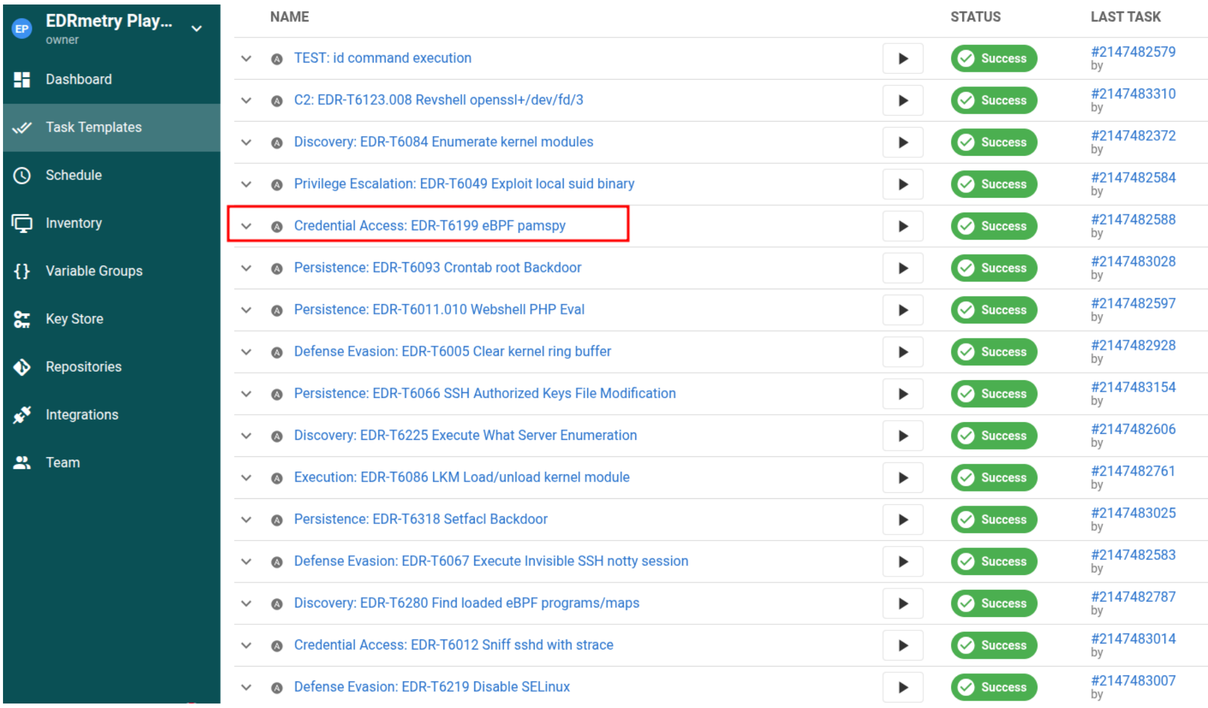
2. Navigate to the EDRmetry Matrix.
3. Choose Tactic and search Technique.

4. Pick a technique EDR-ID (e.g., eBPF pamspy)



Command and Control 16 techniques	Credential Access 14 techniques	Defense Evasion 79 techniques	Discovery 23 techniques	Execution 20 techniques	Exfiltration 17 techniques	Impact 6 techniques	Initial Access 17 techniques	Lateral Movement 18 techniques	Persistence 55 techniques
C2 Implants (8)	Dump credentials via unshadow	ASM Injection over /proc/PID/mem	/proc/PID/ Enumeration	Bash HTTP GET data with /dev/tcp	DNS Exfiltration with dig	Bash Fork Bomb	ActiveMQ CVE-2023-46604 Exploitation	Active Directory Pentesting using Linux	/etc/modules-load.d Persistence
DNS AXFR Payload Delivery	Dump heap memory from Java	Avoid Filename and Filepath Matching	C2 randomized hostname lookups	Built-in System Tools Execution	DNS Tunneling/Exfiltration with dnscat2	Clear kernel ring buffer	Apache HTTP CVE-2021-41773 Exploitation	Create a SOCKS proxy with ssh	/etc/sudoers Modification
DNS Tunneling with iodine	eBPF bcc Sniffs pam_get_authtok() with python3	Bash Anti-Forensic Log Wiper	Check ASLR configuration	Dump process memory via GDB	eBPF Magic String Tracepoint Execution with bpftrace	Crypto Mining CPU stress	Apache Tomcat Manager Exploitation	DarkFlare TCP over CDN Tunneling	Add Backdoor User - /etc/passwd modification
eBPF Keylogger + DNS RCE	eBPF Capture TLS/SSL functions with Qtap	Bash Script Obfuscation	Check bpf settings from /proc	eBPF system("whoami") Execution with bpftrace	Exfil data using rsync	Ransomware bash+openssl	Code Execution via SSH XZBackdoor	DNS Zone Transfer	Add backdoor user with uid=0
Emp3r0r C2 Shadowsocks C2	eBPF pamspy	Bashrc File Hiding with ls Alias	Dismap Asset Discovery	Execute binary listening from a hidden directory as root	Exfil data using transfer.sh	Ransomware Black Basta	HTTPD CVE-2014-6271 Shellshock RCE	Execute Port Scanning	Add new group
Execute process via ProxyChains	eBPF Sniff pam_get_authtok() with bpftrace	Binary Runtime Crypter in Bash	Download and launch LinEnum	Execute LKM call_usermodehelper() on ICMP	ICMP Python Scapy Exfiltration	Ransomware C - lokpack	JetBrains TeamCity CVE-2023-42793	Execute SSHD as a victim user	Add User to Privileged Group
Fileless Reverse shell with sshx	eBPF Sniff PTY with bpftrace	Block rsyslogd logging	Enumerate kernel modules	Execute mknod/mkfifo	ICMP_exfil + nping Exfiltration		K8S - Kubeconfig file	FRP Fast Reverse Proxy	At job persistence
Make Non-standard port HTTP/HTTPS connection	eBPF Sniff SSL/TLS Traffic	BOF Loading with BOF-Stager	Execute "What Server" Enumeration	Export proxy_http	NTP Data Exfiltration		Kafka CVE-2023-25194 Exploitation	Get malicious samples from MalwareBazaar	Backdooring Initramfs
Ngrok Tunneling	Find local passwords/secrets	Bypassing libc hooks with io_uring	Execute LinPEAS from /dev/tcp	File Transfer to a hidden directory	PAM creds over HTTP Post		MySQL Brute Force	Hijack SSH Client Session	BDS Ftrace Hooking Rootkit
Reverse DNS Tunnel Backdoor	K8S - Dump etcd database	Change Shell Optional Behavior	Execute nping	Install suspicious RPM package	pam_exec SSHD Exfiltration		Ofbiz CVE-2024-45507 SSRF+RCE	Ligolo-ng Reverse TCP/TLS Tunneling	Cap_setuid over LD linker
Reverse shells (17)	K8S - Steal Pod Service Account Token	Clear from /var/log/secure	Find all suid/sgid files	K8S - Sidecar injection	Python FTP Upload		OpenSMTPD CVE-2020-7247 RCE	Network ping sweep	Crontab root Backdoor
Shell Over Reverse SSH	Read /etc/shadow	Clear kernel ring buffer	Find all writeable dirs	LKM Load/unload kernel module	SMB Data Exfiltration with impacket		Oracle WebLogic SSRF Exploitation	Proxychains TOR connection	Deploy Malicious Docker Container
SOA/ECS DNS C2 Channel	Scan bash_history to find pass/API keys	Clear Paging Cache	Find loaded eBPF programs/maps	Modify core_pattern file	Telegram Data Exfiltration		Remote UAF Exploitation - root	Reverse SOCKS5 proxy	Deploy malicious RPM package
SSH-based Reverse Shell from NHAS	Sniff sshd with strace	Copy/rename commands to exotic directory	Find SSH keys	MySQL UDF Command Execution	Upload data over HTTP/HTTPS		Remote UAF Exploitation - user	Socks Proxy from Tomcat JSP	DNF Package Manager
Upgrade a reverse shell to a PTY shell		Create file with Unicode zero-width space	Get Kernel Text Region Address	OpenSSL - hackshell download without curl	Upload data over SCP/SFTP		Solr Log4J JNDI Exploitation	SSH Linux Tunneling	eBPF Boopkit Rootkit
XOR shell_reverse_tcp Loader		Curing - io_uring rootkit	Kcore Memory File Read	Perl - File download without curl	Upload data over WebDAV		Spring CVE-2022-22963 Exploitation	SSHD Manipulation in sshd_config.d	eBPF Magic SRC Port Tracepoint Exe with bpftrace
		Disable .bash_history	Linux VM Check via Hardware		Upload/download data over SSHFS			Tailscale Tunneling	eBPF mount bpffs
		Disable ASLR	Linux VM Check via Kernel Modules					Visit malicious Threat Intel URL	eBPF sudo Rootkit

5. Navigate to the EDRmetry Pulse Dashboard.
6. Find a corresponding EDR-ID within the chosen Tactic:

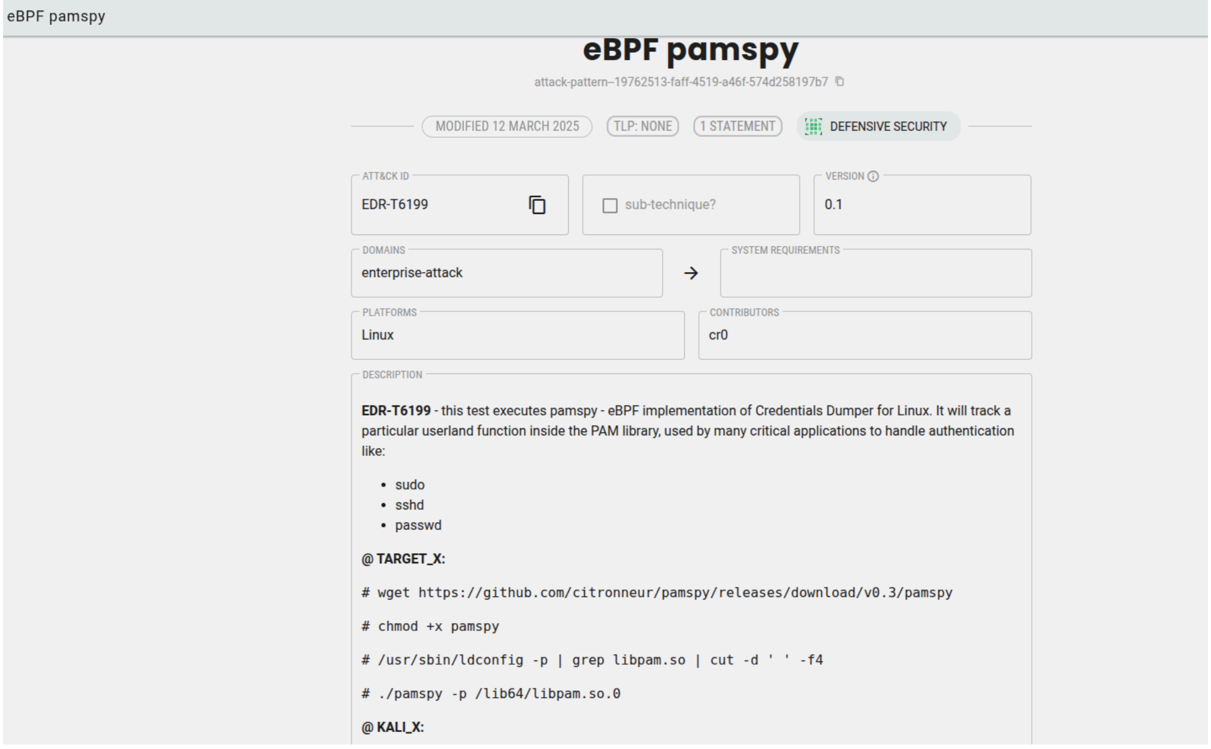


NAME		STATUS	LAST TASK
TEST: id command execution	▶	✓ Success	#2147482579 by
C2: EDR-T6123.008 Revshell openssl+/dev/fd/3	▶	✓ Success	#2147483310 by
Discovery: EDR-T6084 Enumerate kernel modules	▶	✓ Success	#2147482372 by
Privilege Escalation: EDR-T6049 Exploit local suid binary	▶	✓ Success	#2147482584 by
Credential Access: EDR-T6199 eBPF pamspy	▶	✓ Success	#2147482588 by
Persistence: EDR-T6093 Crontab root Backdoor	▶	✓ Success	#2147483028 by
Persistence: EDR-T6011.010 Webshell PHP Eval	▶	✓ Success	#2147482597 by
Defense Evasion: EDR-T6005 Clear kernel ring buffer	▶	✓ Success	#2147482928 by
Persistence: EDR-T6066 SSH Authorized Keys File Modification	▶	✓ Success	#2147483154 by
Discovery: EDR-T6225 Execute What Server Enumeration	▶	✓ Success	#2147482606 by
Execution: EDR-T6086 LKM Load/unload kernel module	▶	✓ Success	#2147482761 by
Persistence: EDR-T6318 Setfacl Backdoor	▶	✓ Success	#2147483025 by
Defense Evasion: EDR-T6067 Execute Invisible SSH notty session	▶	✓ Success	#2147482583 by
Discovery: EDR-T6280 Find loaded eBPF programs/maps	▶	✓ Success	#2147482787 by
Credential Access: EDR-T6012 Sniff sshd with strace	▶	✓ Success	#2147483014 by
Defense Evasion: EDR-T6219 Disable SELinux	▶	✓ Success	#2147483007 by

7. Hit the "Play" button.
8. Check execution status.
9. Verify detections and alerts → Check telemetry, detections, and alerts generated within the chosen EDR/Runtime/SIEM platform.

10. Adjust detection logic if necessary or ask questions to the EDR/SIEM vendor.

11. Learn more about the chosen EDR-ID technique.



eBPF pamspy

eBPF pamspy

attack-pattern--19762513-faff-4519-a46f-574d258197b7

MODIFIED 12 MARCH 2025 | TLP: NONE | 1 STATEMENT | DEFENSIVE SECURITY

ATT&CK ID: EDR-T6199

□ sub-technique?

VERSION: 0.1

DOMAINS: enterprise-attack

SYSTEM REQUIREMENTS:

PLATFORMS: Linux

CONTRIBUTORS: cr0

DESCRIPTION

EDR-T6199 - this test executes pamspy - eBPF implementation of Credentials Dumper for Linux. It will track a particular userland function inside the PAM library, used by many critical applications to handle authentication like:

- sudo
- sshd
- passwd

@ TARGET_X:

```
# wget https://github.com/citronneur/pamspy/releases/download/v0.3/pamspy
```

```
# chmod +x pamspy
```

```
# /usr/sbin/ldconfig -p | grep libpam.so | cut -d ' ' -f4
```

```
# ./pamspy -p /lib64/libpam.so.0
```

@ KALI_X:

Target Audience

EDRmetry Pulse is tailored for Cyber Security Professionals, with maximum value for:

- SIEM/EDR Linux Specialists
- Detection Engineers
- SOC Team Members
- Blue Team Defenders
- Purple Team Operators
- Red Team Operators
- SecOps / DevSecOps Engineers
- Threat Hunters
- General Cyber Security Analysts
- Linux Experts
- EDR/Runtime Security Vendors

Use cases → Better Blue by playing Red

- Understand the Linux threat ecosystem and the corresponding offensive techniques in the simplest, automated way, reducing boring, manual effort

- Proactively validate whether the chosen Runtime Security or EDR/XDR engine generates logs, detections, and alerts when a specific technique is executed.

- Identify SIEM blind spots and enhance detection rules, telemetry pipelines, and data source correlations by pinpointing the areas targeted by threat actors.

- Improve your Incident Response capabilities by using EDRmetry Pulse as a basis for internal purple team exercises (red vs blue team)

- Automate and chain offensive techniques to simulate real-world Linux attack scenarios in an active, ongoing process as a part of a detection engineering effort

- Focus on detection engineering and increase your threat hunting capabilities while maintaining the active defense approach

- Find corresponding forensics TTPs artifacts and know better Linux internals

- Find criteria and features to consider when evaluating a Linux EDR platform, and be able to ask Linux EDR/SIEM vendors the right questions about their products

---

CISO/ Board Management Perspective:

- **How does EDRmetry contribute to the overall cybersecurity strategy?**
  - Proactive Defense: Enables organizations to stay ahead of potential threats by understanding and testing against the latest attack techniques
  - Informed Decision Making: Provides concrete data to support EDR/XDR selection and optimization
  - Skill Development: Enhances the capabilities of internal security teams through practical experience
  - Compliance Support: Helps in demonstrating due diligence in security testing and improvement efforts
  - Cost Efficiency: Reduces the need for multiple tools or extensive external consultations for Linux security testing

- **What specific benefits does EDRmetry offer to CISOs and Security Directors?**
  - Comprehensive Visibility: Gain a clear understanding of your Linux environment's security posture

    - Resource Optimization: Make informed decisions about security investments based on actual performance data
    - Risk Management: Identify and address security gaps before they can be exploited
    - Team Empowerment: Provide your security team with advanced tools to enhance their skills and effectiveness
    - Vendor Management: Improve negotiations with EDR/XDR vendors by having concrete data on product performance

- **How does EDRmetry support compliance and audit requirements?**
    - Evidence Generation: Creates detailed logs of security tests and their outcomes
    - Gap Analysis: Helps identify areas where security controls may be insufficient for compliance requirements
    - Continuous Improvement: Supports ongoing security posture assessment and enhancement
    - Documentation: Provides materials that can be used to demonstrate security testing efforts to auditors

- **How can organizations measure the ROI of implementing EDRmetry?**
    - Detection Improvement: Quantify the increase in threat detection rates
    - False Positive Reduction: Measure the decrease in false alarms after optimizing EDR/XDR configurations
    - Incident Response Efficiency: Track improvements in response times and effectiveness
    - Training Cost Reduction: Calculate savings from in-house skill development vs. external training
    - Breach Prevention: Estimate potential cost savings from preventing security breaches

- **What performance metrics can be tracked using EDRmetry?**
    - Detection Coverage: Percentage of known attack techniques successfully detected
    - Time to Detection: Average time taken to identify malicious activities
    - False Positive Rate: Number of false alarms generated during testing
    - Evasion Success Rate: Percentage of techniques that successfully evade detection
    - System Impact: Performance impact of security solutions under various attack scenarios

Summary

EDRmetry Pulse is your ultimate companion in mastering the Linux threat landscape through offensive security automation. Designed with defenders in mind, it bridges the gap between red and blue teams by emulating real-world Linux attacks in a controlled, systematic, and intelligent way—so you can stop threats before they escalate:

Whether you're validating detection efficacy, training your team, or leveling up your security stack, **EDRmetry Pulse** gives you the clarity, automation, and depth needed to stay ahead of evolving Linux threats.

Leszek Mis @ Defensive Security

Security Researcher/CEO at **Defensive-Security.com**, providing open-source cybersecurity services including Linux-oriented Red Team adversary emulations, Blue Team detection coverage testing, EDR effectiveness validation, Incident/DFIR support. Trainer at Black Hat USA/Asia, Hack In The Box Abu Dhabi/Singapore/Amsterdam, OrangeCON, x33fcon. Providing live workshops and high-quality knowledge transfers. Over 20 years of hands-on experience in Linux Red/Blue. My areas of interest include the development of multi-stage attack paths mapping to MITRE ATT&CK, multi-level detection paths known as detection engineering, Linux/network-related ML feature extraction, Linux internals with a focus on kernel-space/eBPF rootkits, Detection Engineering, deep log/memory analysis, threat hunting, and exploration of new offensive techniques in Linux/Kubernetes vs DFIR/detection and protection/hardening techniques. Red Hat Certified Architect (RHCA), OSCP, Splunk Architect. Creator of PurpleLabs Cyber Range and author of a widely recognized Linux Attack, Detection, and Live Forensics course. Learning hard every single day.

- LinkedIn: https://www.linkedin.com/in/crony/
- Twitter/X: https://x.com/cr0nym

Company Details - Defensive Security

Defensive Security Holding Sp. z o.o.
ul. Kamienna 1e
Wilkowice, 43-365, Poland
VAT EU: PL9372756070
https://defensive-security.com

DEFENSIVE
SECURITY