Advanced Linux Detection and Forensics CheatSheet by Defensive Security v0.4 [10/09/2024]

DEFENSIVE SECURITY

/proc:

/proc/modules → Displays a list of all modules loaded into the kernel

/proc/kallsyms → Displays addresses of kernel symbols

/proc/vmallocinfo → Gives mapping of virtual address space of the kernel

/proc/PID/maps → Lists of all the memory-mapped files of a process

/proc/PID/maps | grep '(deleted)' → Lists of deleted memory-mapped files of a process (ex. deleted shared libraries)

/proc/PID/fd/* → Get file descriptors per process

/proc/PID/fd/* | grep 'memfd' → Get processes with anonymous (memory-backed) file descriptors live in RAM

/proc/PID/fdinfo → Contains one entry for each file that the process has open

/proc/PID/map_files/* → Contains entries corresponding to memory-mapped files

/proc/PID/environ → Display environment variables per process

/proc/PID/exe → A symbolic link containing the actual pathname of the executed command

/proc/PID/exe | grep 'deleted' → A symbolic link containing the actual unlinked pathname of the executed command

/proc/PID/comm → Exposes the process's comm value - that is, the command name associated with the process

/proc/PID/cmdline → Holds the complete command line for the process

/proc/PID/cwd → Gets a symbolic link to the current working directory of the process

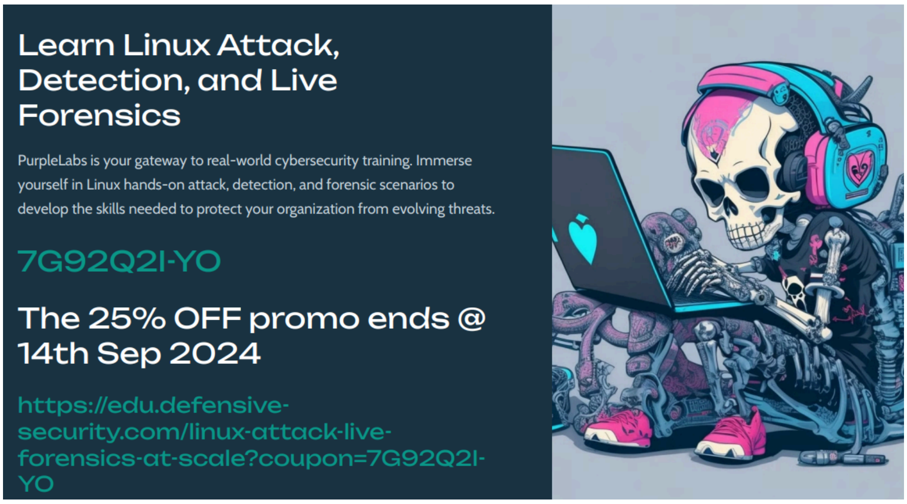
/proc/PID/status → Status information about the process used by ps

/proc/PID/stack → Symbolic trace of the function calls in this process's kernel stack

/proc/PID/stack | grep packet_recvmsg
/proc/PID/stack | grep wait_for_more_packets → get processes with packet capture functions

/proc/net/unix → List UNIX sockets

/proc/net/nf_conntrack → records the source IP, destination IP, and other information of a TCP connection in the ESTABLISHED state



/proc/mounts → Lists of all the filesystems currently mounted on the system

/proc/PID/fd/* | grep bpf-map → Get file descriptors per process with bpf-map type

/proc/PID/fd/* | grep bpf-prog → Get file descriptors per process with bpf-prog type

/proc/sys/kernel/tainted → Display the kernel-tainted state

/proc/PID/task/TID/children → Space-separated list of child tasks of this task

/sys:

/sys/kernel/debug/tracing/enabled_functions → contains a list of kernel functions that are currently enabled for tracing

/sys/kernel/debug/tracing/trace → Get trace events

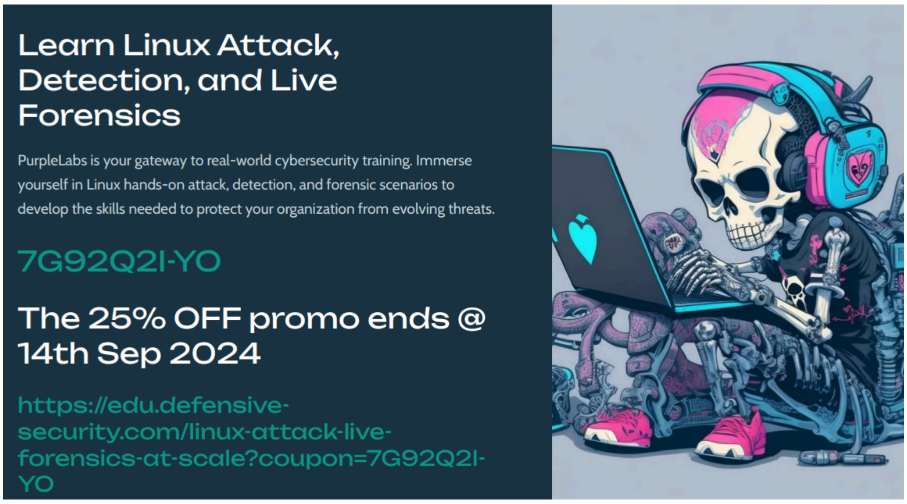
/sys/kernel/tracing/available_filter_functions → Provides a list of available functions that you can use as filters when setting up tracing

/sys/module/* → List loaded kernel modules, and compare with /proc/modules

/sys/module/$module/parameters → Check available parameters per module

/sys/module/$module/taint → Indicates whether a loaded kernel module has "tainted" the kernel

/sys/fs/bpf/* → List pinned eBPF progs



Logs:

/var/log/messages → Contains global system messages, including the messages that are logged during system startup

/var/log/auth.log → Authentication logs

/var/log/kern.log → Kernel information and events

/var/log/secure → Authentication logs

/var/log/syslog → Contains messages that are recorded by the host about the system activity

/var/log/httpd/ → Apache logs

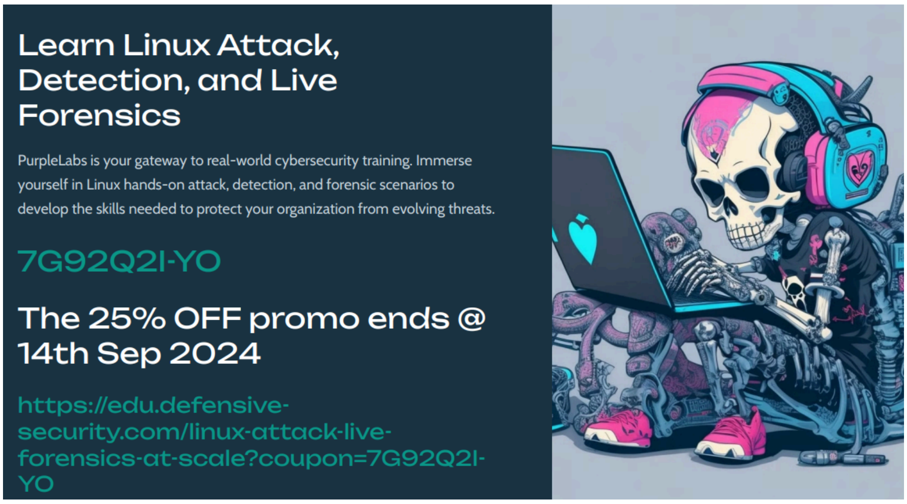
/var/log/daemon.log → Contains information about running system and application daemons

/var/log/cron → Cron logs

/var/log/auditd/audit.log | grep denied → Get SELinux alerts

/var/log/journal → journald systemd's logs

journalctl --file X.journal -o verbose > journal.txt → Dump journald logs with verbose output



CLI/tools:

lsmod → Display the status of modules in the Linux Kernel by reading /proc/modules

lsof → "list open files" tool is a robust interface for the information inside the /proc virtual filesystem

ls -al → find hidden files

env → Display environment variables

who / w / pinky → Show logged users

last → show a listing of the last logged-in users based on /var/log/wtmp

lastb → Show a listing of the last unsuccessful logins based on /var/log/btmp

ps -efwwww → Get a full list of running processes

grep . FILENAME → single byte read to decloak the file

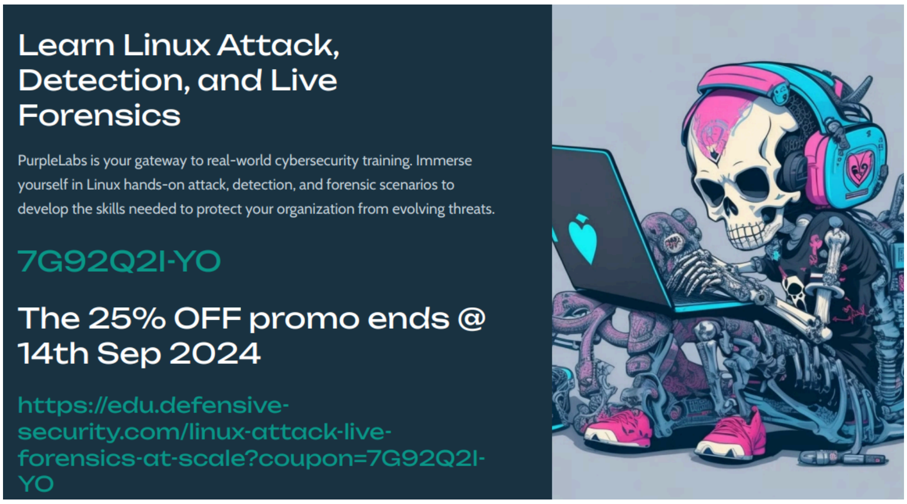
pstree → Display a tree of processes

find → Find files and directories

dd if=mem bs=1 skip=ADDRESS count=1000 of=/tmp/dumped_proc_file → Extract memory content (1000 bytes) at specified ADDRESS

service --status-all → Display System V services status information

stat → Display file or file system status



readelf →Display information about ELF files

objdump → Display information from object files

strings → Determines the contents of non-text files

capa → Tool to identify capabilities in executable files

yara → Identify and classify malware samples

strace → Trace system calls and signals

ltrace → intercepts and records the dynamic library calls which are called by the executed process and the signals which are received by that process

ip link show | grep xdp →Find if any of network interfaces have XDP enabled

ip link show | grep qdisc →Find if any of network interfaces have Traffic Control enabled

sudoreplay → Replay sudo session logs

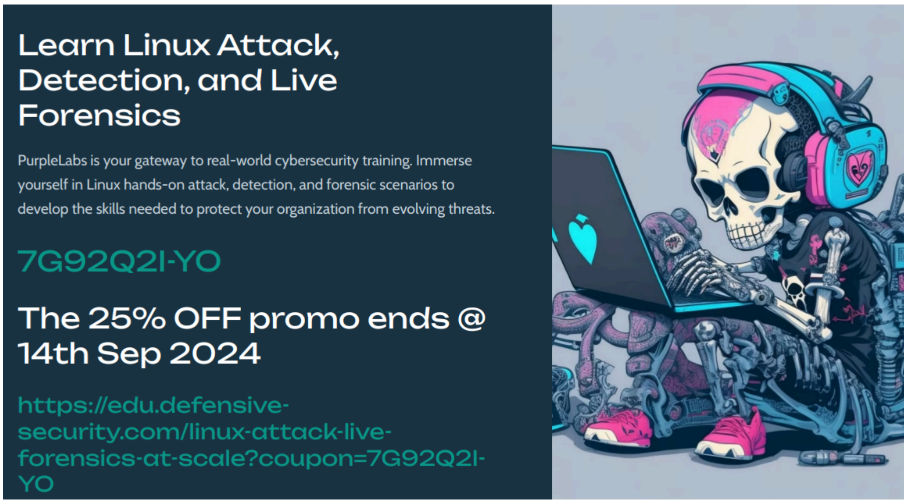
bpftool prog list → List loaded eBPF programs

bpftool map list → List eBPF maps

dmesg | grep bpf_probe_write_user → Check for the presence of bpf 'bpf_probe_write_user' helper

dmesg | grep taint → Check kernel message buffer for tainted kernel modules

dmesg | grep systemtap → Check for the presense of systemtap



mount → Read /proc/mounts, watch for bind-mounted PID dirs to random dir

top → Display current running processes

iptables -L -v -n → Collect firewall rules

iptables -t nat -L -v -n → Collect firewall rules from nat chain

ss → Display listening sockets

uptime → Display how long system has been running

auditctl -l → Display kernel's audit rules

ausearch → Query the audit daemon logs for events based on different search criteria

chkconfig --list → Display a list of all services and their current configuration

systemctl list-units → Display all systemd system units

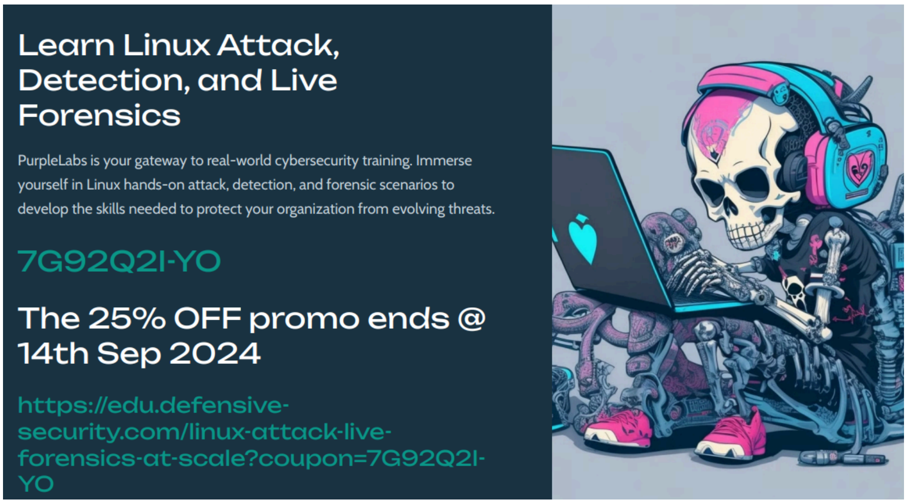
systemctl list-timers --all → Display timer units currently in memory

systemctl list-unit-files → Display unit files installed on the system

loginctl user-status UID --full → May be used to introspect and control the state of the systemd login manager per user

getenforce → Display the current mode of SELinux

sestatus -v → Display the contexts of files and processes listed in the /etc/sestatus.conf file



dnf list installed → Display installed packages

yum list installed → Display installed packages

dpkg -l → Display installed packages

rpm -V -a → Verify all packages to compare information about the installed files in the package with information about the files taken from the package metadata stored in the rpm database

debsums → Verify installed Debian package files against MD5 checksum lists from /var/lib/dpkg/info/*.md5sums

tc qdisc → show/manipulate traffic control settings

ext4magic → List/recover deleted files

log2timeline.py → extract events from individual files and creates a Plaso storage file

getcap -r / 2>/dev/null → displays the name and capabilities of each file

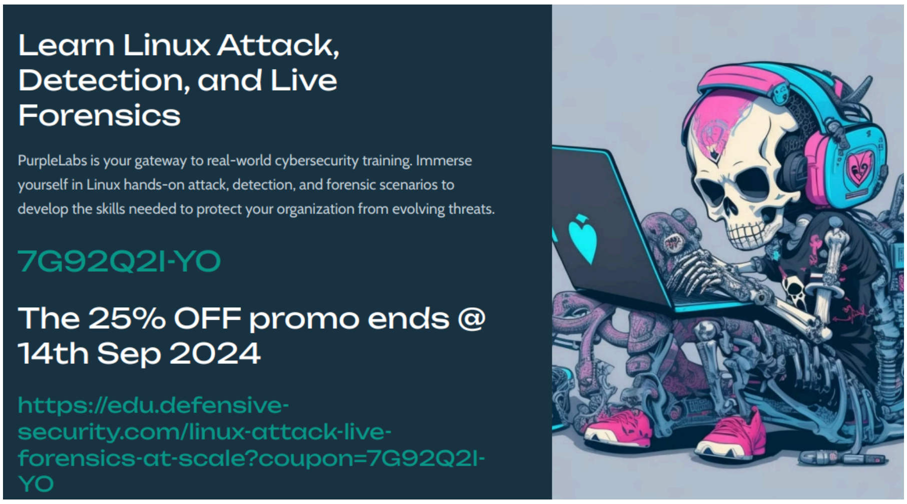
BPFhookdetect → Detect syscall hooking using eBPF

inotify → Provides a mechanism for monitoring filesystem events

lsattr → List file attribute ex. immutable bit

base64 → Encode/decode data and print to standard output

LKRG → Performs runtime integrity checking of the Linux kernel and detection of security vulnerability exploits against the kernel



https://edu.defensive-security.com/linux-attack-live-forensics-at-scale?coupon=7G92Q2I-YO

Files/directories/attributes:

.bash_history → Get the command history for the Bash shell

.mysql_history → Get the query history for the MySQL/MariaDB sessions

.ftp_history → Get the command history for the FTP (File Transfer Protocol) client

.git/logs → Get log files that track changes to the repository's references and branches

/etc/passwd → Get essential information about user accounts

/etc/group → Get essential information about user groups

/etc/fstab → Contains descriptive information about the filesystems the system can mount

/etc/ssh/sshd_config → Main sshd configuration file

/etc/sudoers → Contains default sudo security policy configuration

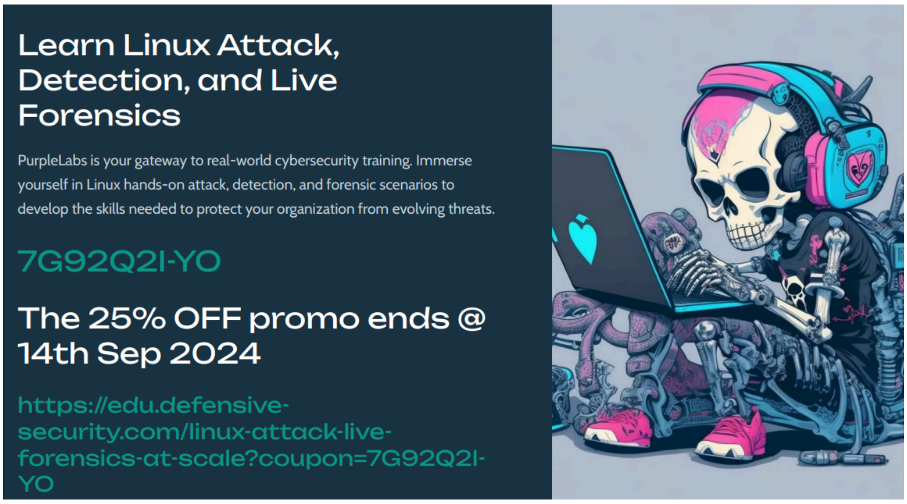
.ssh/authorized_keys → Get a list of public SSH keys that are authorized to access the user's account

.ssh/known_hosts → Stores information about the public keys of remote SSH servers

.viminfo → Get various types of information between editing sessions

.gitconfig → Get settings and preferences for Git repositories and user accounts

/boot/initrd.img → contains the necessary executables and system files to support boot of a Linux system



/etc/ld.so.preload → Contains a whitespace-separated list of ELF shared objects to be loaded before the program

/lib64/ld-2.X.so → Dynamic linker which finds and loads the shared objects (shared libraries) needed by a program

/dev/shm/ → shared memory implementation

/dev/ → List device files and directories

suid → Search for files that have SUID bit set

sgid → Search for files that have SGID bit set

/etc/cron* /var/cron* /etc/at* → Linux scheduler

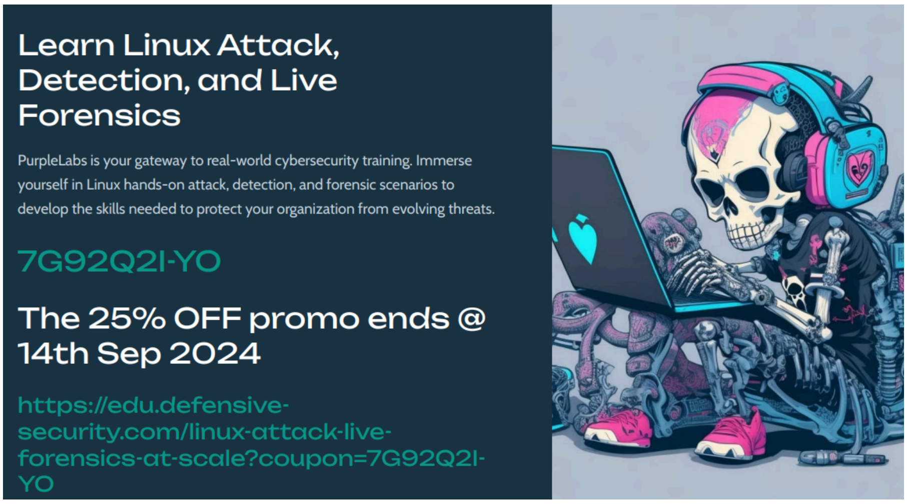
/etc/pam.d → main PAM configuration files

OSquery/Sunlight/osquery-defense-kit → OSquery queries for Detection & Incident Response:

deleted-or-replaced.sh → Reveal processes that are powered by deleted programs

maps-deleted.sh → Detect processes with loaded deleted shared libraries within memory address space

fake-name.sh → Uncover unexpected programs that are faking their names



hidden-files.sh → Reveal hidden files

hidden-parent-pid.sh → Find processes that have hidden parent IDs

hidden-pids.sh → Reveal rootkits that hide processes from getdents() calls to /proc

hidden-pids-mount.sh → Detect potential malicious behavior that hides processes from ps using mount -o bind

pid-hidden-by-rootkit.sh → Finds processes that are apparently hidden by a rootkit

hidden-sys-module.sh → Reveal if there is a hidden /sys/module entry

kernel-taint.sh → Diagnose tainted kernels

ld-so-preload.sh → Find preload entries

mystery-char-devices.sh → Uncover mysterious character devices in /dev

raw-packet-sniffer.sh → Detect raw socket sniffers

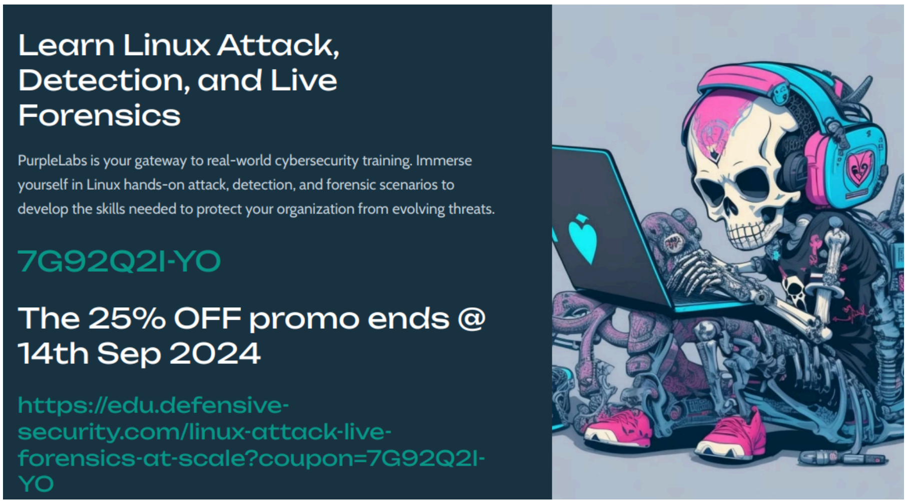
rootkit-signal-handler.sh → Detect rootkits, such as Diamorphine, that respond to exotic signals

root-socket-no-libraries.sh → Reveal processes running as root with a socket but no dependencies outside of libc

root-ssh-authorized-keys.sh → Find root SSH authorized keys

suspicious-cron.sh → Reveal suspicious crontab entries

suspicious-proc-env.sh → Find processes that have unusual environment variables



thieves.sh → Reveal programs whose process space may have been taken over by another program

unexpected-ebpf-hooks.sh → Discover suspicious behavior in eBPF

unexpected-run-locks.sh → Reveal processes with weird lock files open in /var/run

unexpected-trace-pipe.sh → Discover kernel modules logging to the trace pipe - this may be the sign of an eBPF-based rootkits

world-readable-run-locks.sh → Show world readable locks in /var/run

bpf-find-maps.sh → Find suspicious bpf maps

bpf-find-progs.sh → Find suspicious bpf programs

bpf-probe-write-user.sh → Find suspicious bpf write user in dmesg

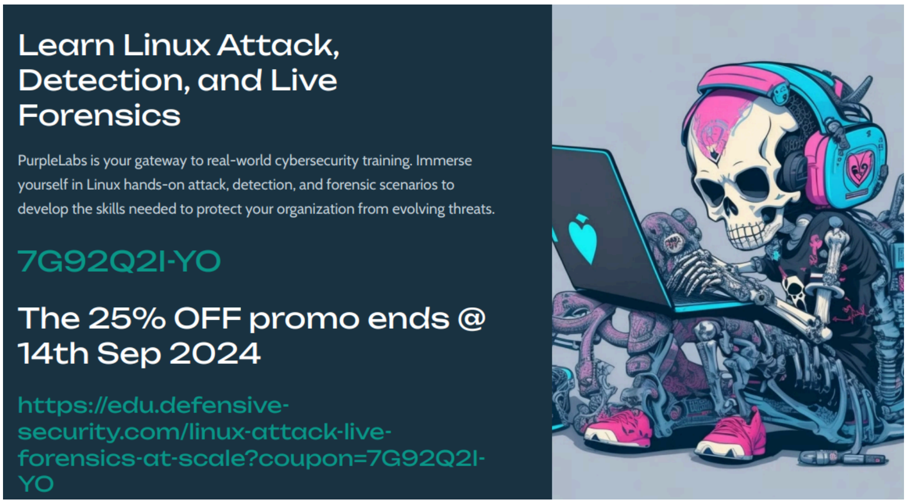
unexpected-ebpf-hooks.sh → Detect suspicious bpf hooks

overwritten-memory-map-ddexec-linux.sh → Detect processes with a memory map that suggests they might be code smuggling

listening-from-unusual-location.sh → Find unexpected programs listening from /tmp or other weird directories

low-fd-socket.sh → Find programs where fd0 (stdin), fd1 (stdout), or fd2 (stderr) are connected to a socket

reverse-shell-socket.sh → Detect potentially suspicious reverse-shell processes



unexpected-dns-traffic.sh → Catch DNS traffic going to machines other than the host-configured DNS server

unexpected-etc-executables.sh → Find unexpected executable files in /etc

unexpected-etc-hosts.sh → Find unexpected potentially suspicious /etc/hosts entries

unexpected-privilege-escalation_linux.sh → Find processes that run with a lower effective UID than their parent PID

unexpected-shell-parents.sh → Find unexpected process that spawns shell processes

unexpected-talkers-linux.sh → Find unexpected programs communicating over non-HTTPS protocols

unusual-process-name-linux.sh → Find processes with suspicious executable names

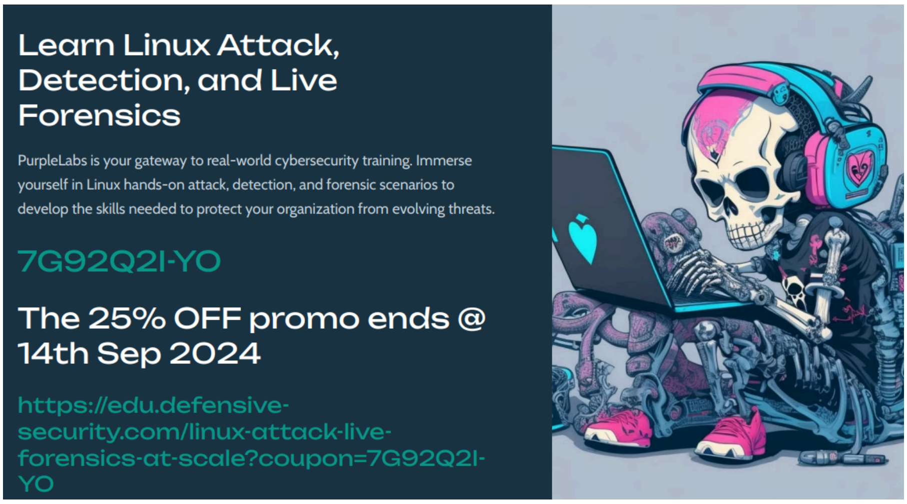
unexpected-dev-entries.sh → Find unexpected files in /dev

unexpected-active-systemd-units.sh → Unexpected systemd units, may be evidence of persistence

unexpected-execdir-linux.sh → Programs running out of unexpected directories

exotic-commands-linux.sh → Find exotic processes based on their command-line

unexpected-privileged-containers.sh → Detect the execution of privileged Docker containers which can be used to escape to the host



unexpected-libcurl-user-linux.sh → Find programs processes which link against libcurl

unexpected-https-linux.sh → Unexpected programs communicating over HTTPS

unexpected-hidden-system-paths.sh → Find unexpected hidden directories in system folders

unexpected-kernel-modules-linux.sh → Find kernel modules that are not part of the expected list

unexpected-setxid-process.sh → Detect running processes that originate from setuid/setgid programs

hidden-modules-filter-functions.sh → Find difference between available_filter_functions and loaded modules

yara-suspicious-strings-process-linux.sh → Find running processes with potentially malicious behavior

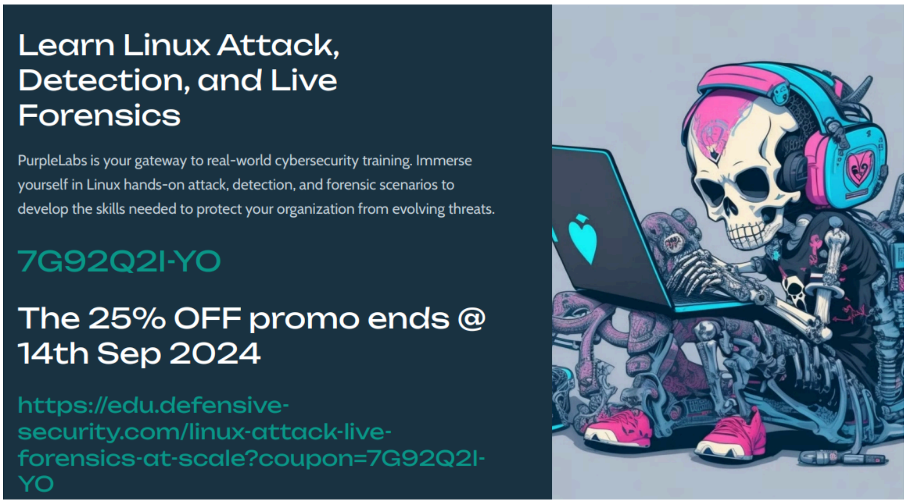
unexpected-var-executables-linux.sh → Find unexpected executables in /var

unexpected-tmp-executables-linux.sh → Find unexpected executables in /tmp

unexpected-dev-executables-linux.sh → Find unexpected executables in /dev

unusual-executable-name-linux.sh → Detect processes with executable names that are potentially suspicious

yara-recently-downloaded-go-crypt-exec.sh → Find running processes with recently downloaded cryptexec behavior



yara-unexpected-upx-process.sh → Find currently running processes backed by UPX executable

unexpected-icmp-socket.sh → Find processes with ICMP socket communication

sudo-preload.sh → Find LD_PRELOAD in /etc/sudoers

sudo.d-preload.sh → Find LD_PRELOAD in /etc/sudoers.d/*

Runtime Security/Tracee→ Linux Runtime Security and Forensics using eBPF:

Anti-Debugging Technique → Detects anti-debugging techniques

ASLR Inspection → Detects ASLR inspections

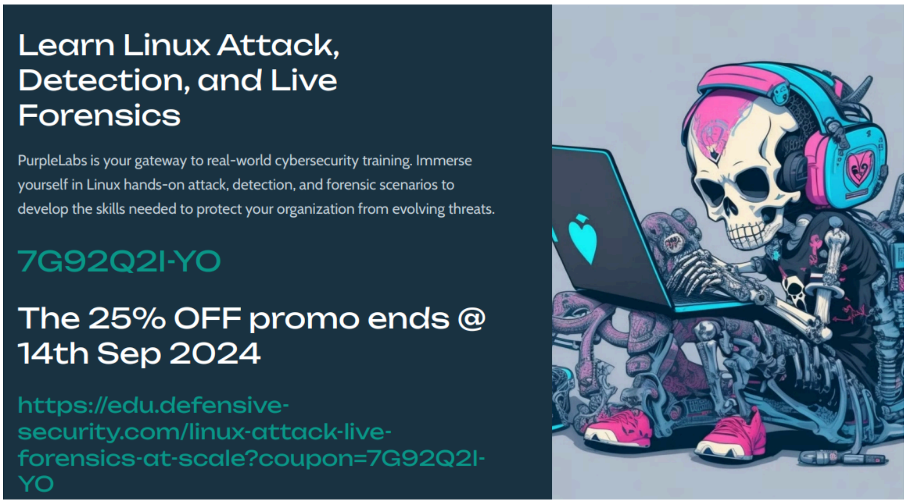
Cgroups notify_on_release File Modification → Monitors notify_on_release file changes in cgroups

Cgroups Release Agent File Modification → Detects changes to the cgroup release_agent

Core Dumps Config File Modification → Monitors core dump configuration alterations.

Default Dynamic Loader Modification → Tracks changes to the default binary loader.

Container Device Mount → Detects unauthorized container device mounts.



Docker Socket Abuse → Flags potential Docker socket misuse

Dropped Executables → Detects runtime-dropped executables.

Dynamic Code Loading → Monitors dynamic code loading events

Fileless Execution → Flags fileless execution techniques

Hidden Executable File Creation → Detects creation of hidden executable files

Illegitimate Shell → Flags unauthorized or unexpected shell executions

Kernel Module Loading → Monitors kernel module load events

Kubernetes API Server Connection → Detects connections to the Kubernetes API server

Kubernetes TLS Certificate Theft → Flags potential theft of Kubernetes certificates

LD_PRELOAD Code Injection → Monitors LD_PRELOAD injection attempts

File Operations Hooking on Proc Filesystem → Detects hooks on file operations in /proc

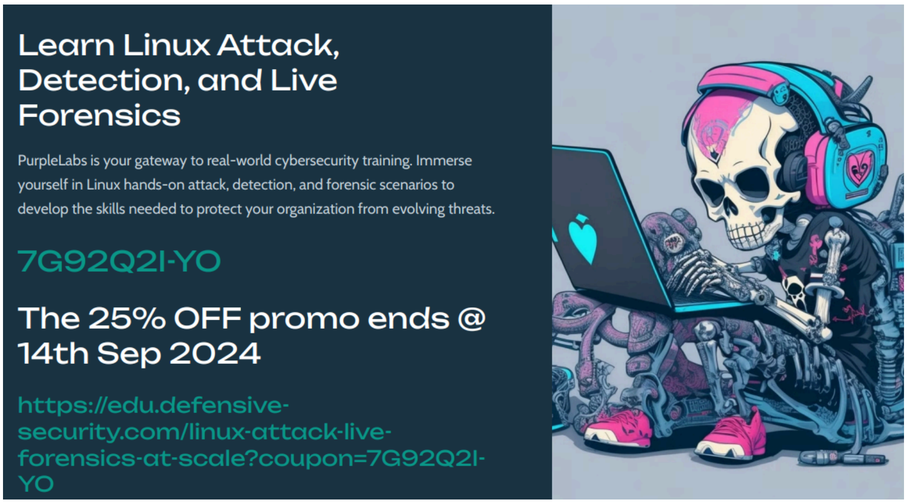
Kcore Memory File Read → Monitors reads of /proc/kcore

Process Memory Access → Flags unauthorized /proc/mem access.

Procfs Mem Code Injection → Detects code injections via /proc/mem

Process VM Write Code Injection → Monitors injections via process_vm_writev

Ptrace Code Injection → Detects ptrace-facilitated code injections.



RCD Modification → Monitors changes to the remote control daemon

Sched Debug Reconnaissance → Flags /proc/sched_debug reconnaissance

Scheduled Tasks Modification → Tracks modifications to scheduled tasks.

Process Standard Input/Output over Socket → Detects IO redirection over sockets

Sudoers File Modification → Monitors alterations to the sudoers file

Syscall Table Hooking → Detects syscall table hook attempts

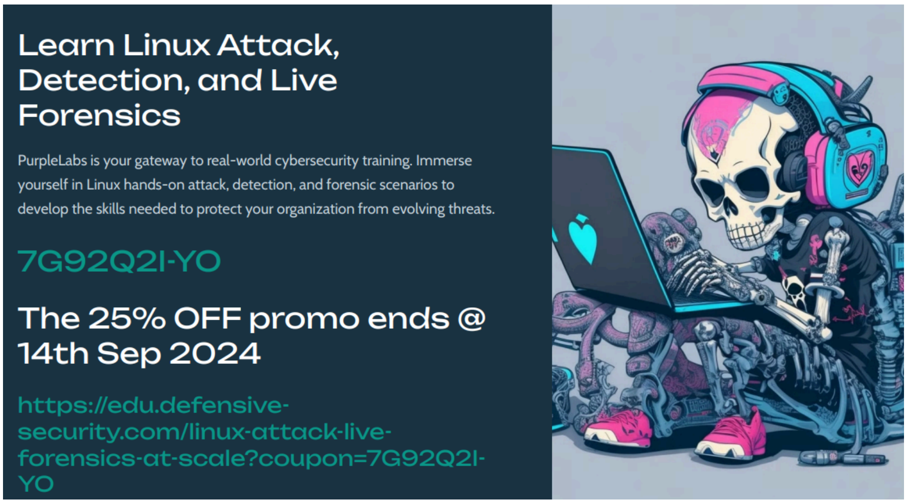
System Request Key Configuration Modification → Monitors system request key configuration changes

Runtime Security/Falco → Detects and alerts on abnormal behavior and potential security threats in real-time:

Disallowed outbound connection destination → Detects any outbound connection to a destination outside of an allowed set of ips, networks, or domain names

Outbound connection to C2 server → Detects outbound connection to command & control servers

Disallowed SSH Connection → Detect any new ssh connection to a host other than those in an allowed group of hosts



Network connection outside authorized port and binary → Detects traffic that is not to authorized server process and port

Possible miner running → Detects crypto miners using the Stratum protocol

File created below /dev by untrusted program →Detects creating any files below /dev other than known programs that manage devices. Some rootkits hide files in /dev.

File created below /etc by untrusted program → Detects creating any files below /etc

File below /etc opened for writing → Detects attempt to write to any file below /etc

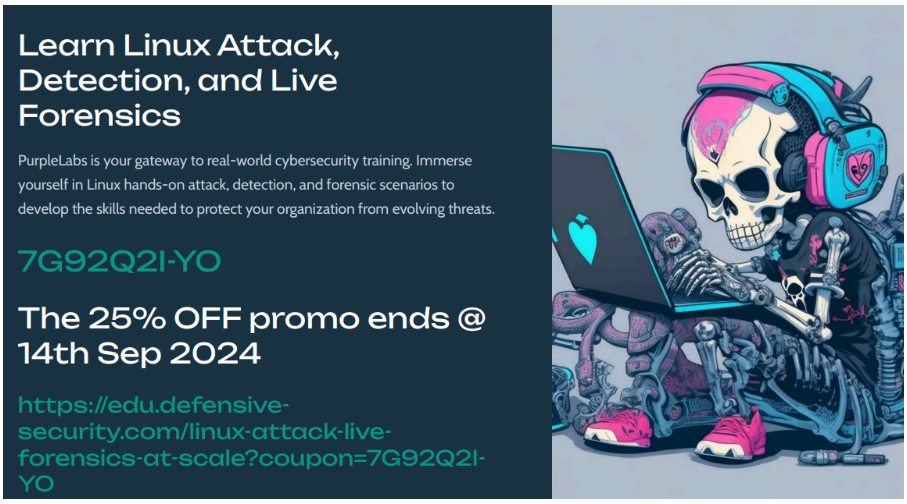
File below / or /root opened for writing → Detects an attempt to write to any file directly below / or /root

Interactive root → Detects anything that runs interactively by root

Privileged container started → Detects the initial process started in a privileged container.

Excessively capable container started → Detects container started with a powerful set of capabilities

Rpm database opened for writing by a non-rpm program → Detects an attempt to write to the rpm database by any non-rpm related program



Shell configuration file has been modified → Detects attempt to modify shell configuration files

Cron jobs were scheduled to run → Detects modifications and executions of cron jobs

Sensitive file opened for reading by non-trusted program → Detects an attempt to read any sensitive file (e.g. files containing user/password/auth info)

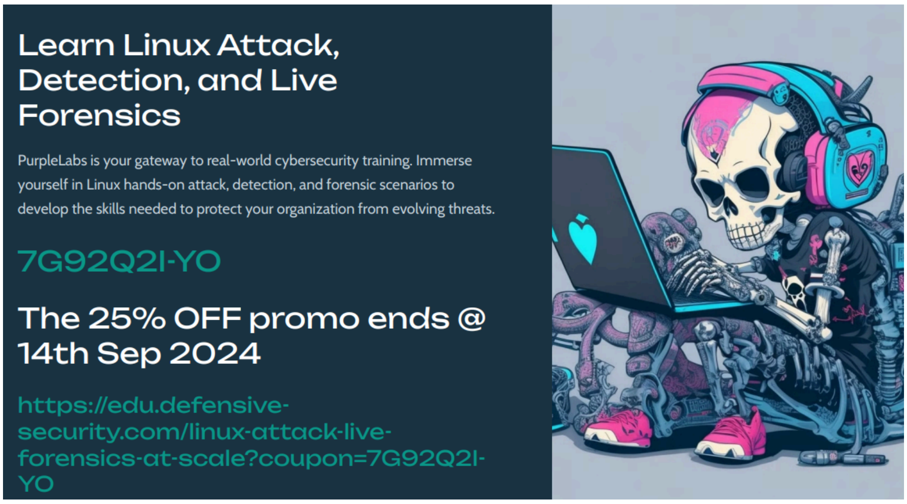
Database-related program spawned process other than itself → Detects a database-server related program spawned a new process other than itself.

Program run with disallowed HTTP_PROXY environment variable → Detects an attempt to run a program with a disallowed HTTP_PROXY environment variable

Known system binary sent/received network traffic → Identifies any network activity performed by system binaries that are not expected to send or receive any network traffic

Redirect stdout/stdin to network connection → Detect redirecting stdout/stdin to network connection

Interpreted program received/listened for network traffic → Detects any inbound network activity performed by any interpreted program (perl, python, ruby, etc.)



Unexpected UDP Traffic Seen → Detects UDP traffic not on port 53 (DNS) or other commonly used ports

Unexpected setuid call by non-sudo, non-root program → Detects an attempt to change users by calling setuid. sudo/su are excluded

Unexpected connection to K8s API Server from container → Detects attempts to contact the K8S API Server from a container

Network tool launched on host → Detects network tools launched on the host

Shell history had been deleted or renamed → Detects bash history deletion

Hidden file or directory created → Detects hidden files or directories created

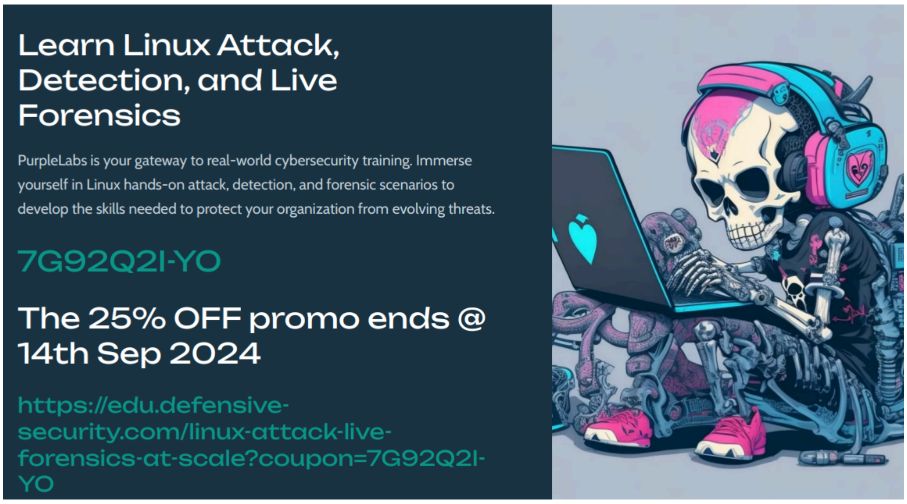
Symlinks created over sensitive files → Detects symlink created over sensitive files

Hardlinks created over sensitive files → Detects hardlink created over sensitive files

An userfaultfd syscall was successfully executed by an unprivileged user → Detects a successful unprivileged userfaultfd syscall which might act as an attack primitive to exploit other bugs

Java process class file download → Detects Java process downloading a class file which could indicate a successful exploit

Outbound connection to IP/Port flagged by https://cryptoioc.ch → Detects outbound connections to common miner pool ports



Mount was executed inside a privileged container → Detects file system mount happened inside a privileged container

Detect Sudo Privilege Escalation Exploit (CVE-2021-3156) → Detects Privilege escalation attempt affecting sudo (<= 1.9.5p2)

Linux Kernel Module injection using insmod detected → Detects if kernel module was injected

Detect an attempt to exploit a container escape using release_agent file → Detects an attempt to exploit a container escape using release_agent file

Drift detected (open+create), new executable created in a container → Identifies if new executable created in a container due to open+create

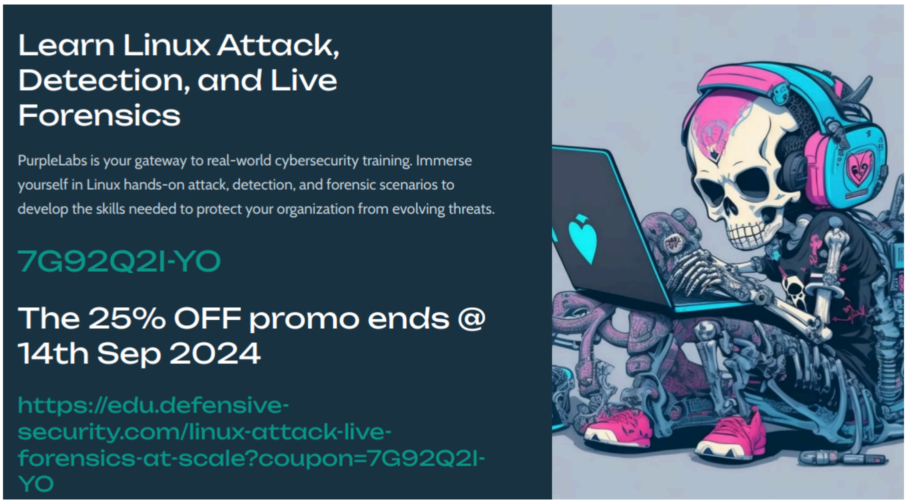
Runtime Security/Kunai → Threat-hunting tool for Linux:

Execve → Generated whenever an execve syscall happens on the system. It provides information about the current binary currently running.

Execve script → Generated under the same conditions as execve event. The only difference is that it provides additional information about the interpreter when the file being executed is a script

Exit → Generated when a single task (process or thread) exits.

Exit group → Generated when a thread-group (process and all its threads) exits



Clone → A task is being cloned/forked. This means that a new task will be created on the system.

Prctl → Generated when a process makes a call to the prctl syscall

Init module → Generated when a kernel module is loaded into the kernel.

Bpf prog load → Generated every time a BPF program is loaded into the kernel.

Bpf Socket Filter Attached → A socket filter attachement has been made

Mprotect exec → Generated when memory protection is turned to executable.

Mmap exec → Generated whenever the mmap syscall is used to map an executable file in memory, with memory execution protection.

Connect → Generated every time a connect attempt is made to a remote IP.

Dns query → Generated when the a DNS response is received on the host and gives insight both on the query, the response and the DNS resolver.

Send data → Generated when data is sent to a remote IP address.

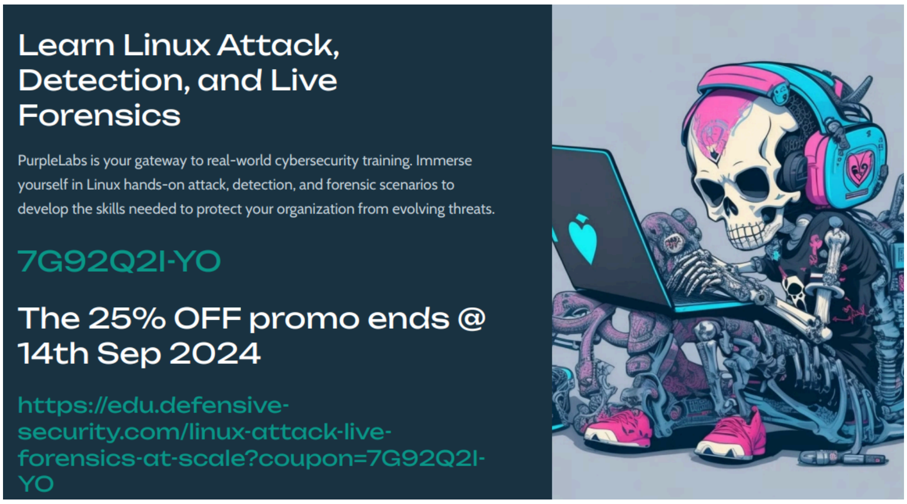
Read → Generated whenever a file is read.

Read config → Generated whenever a file located in /etc is being read

Write → Generated whenever a file is write.

Write config → Generated whenever a file located in /etc is being written.

File rename → Generated whenever a file is being renamed.



File unlink → Generated when a file gets unlinked

Runtime Security/Tetragon → eBPF-based Security Observability and Runtime Enforcement:

Process Lifecycle Monitoring via exec and exit → Mo

Binary Execution in /tmp → Monitors execution of a binary in the /tmp directory.

sudo Monitoring → Monitors sudo invocations

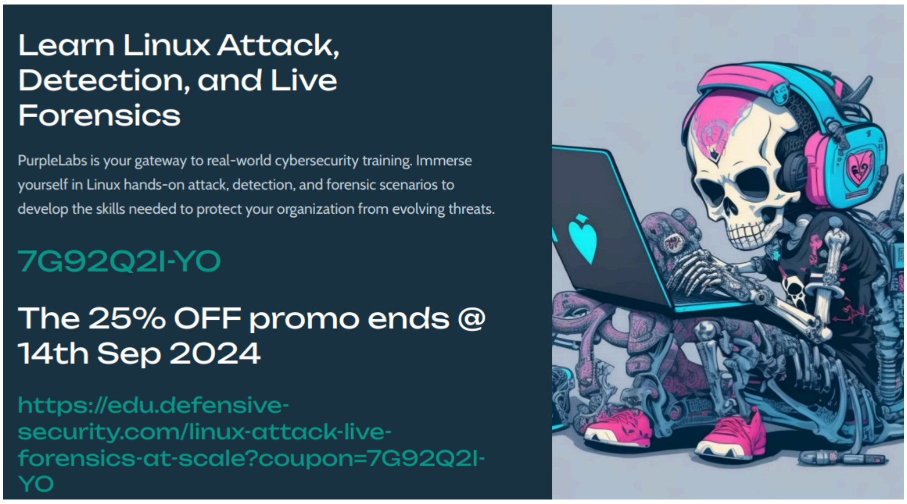
Privileges Escalation via SUID Binary Execution → Monitors execution of SUID "Set User ID" binaries.

Privileges Escalation via File Capabilities Execution → Monitors execution of binaries with file capabilities.

Privileges Escalation via Setuid system calls → Monitors execution of the setuid() system calls family.

Privileges Escalation via Unprivileged User Namespaces → Monitors creation of User namespaces by unprivileged.

Privileges Change via Capset system call → Monitors execution of the capset() system call.



Fileless Execution → Monitors the execution of binaries that exist exclusively as a computer memory-based artifact.

Execution of Deleted Binaries → Monitors the execution of deleted binaries.

eBPF System Activity → Audits BPF program that loads and BPFFS interactions

Kernel Module Audit trail → Audits loading of kernel modules

Shared Library Loading → Monitors loading of libraries

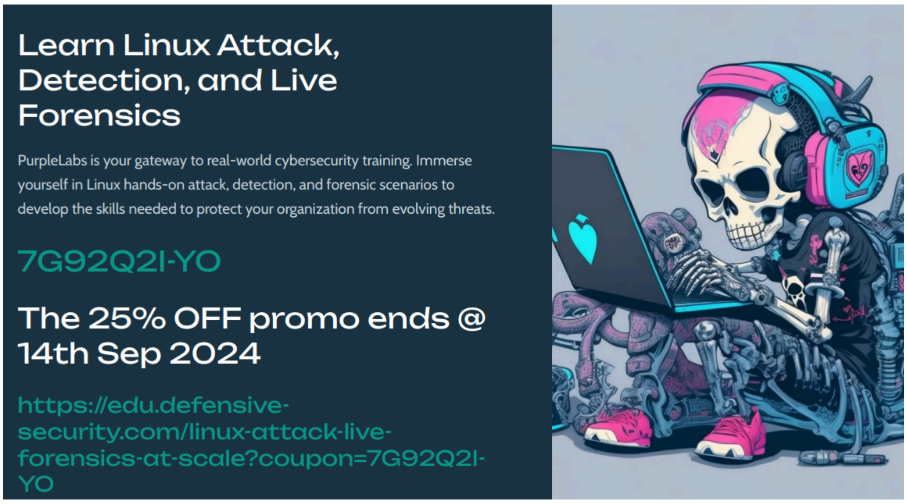
Network Activity of SSH daemon → Monitors sessions established to sshd

Outbound Connections → Monitors all egress connections

Argus → a cutting-edge runtime security tool designed for both monitoring and enforcing application behavior:

capabilities_modification → triggered when there are modifications to the capabilities configuration files in a Linux environment, specifically targeting changes to /etc/security/capability.conf

code_modification_through_procfs → triggered by an attempt to modify code through direct access to process memory via the /proc filesystem



core_pattern_access → triggered when there is an attempt to modify the system's core dump pattern, typically found at /proc/sys/kernel/core_pattern

cpu_fingerprint → Triggered by an attempt to access specific system files that could be used to gather detailed information about the CPU architecture and configuration direct access to process memory via the /proc filesystem

credentials_files_access → Monitors and flags unauthorized or suspicious access to files potentially containing sensitive credentials

filesystem_fingerprint → Triggered when specific system files related to disk and filesystem configurations are accessed ex. /etc/fstab, /proc/diskstats, /proc/filesystems, etc.

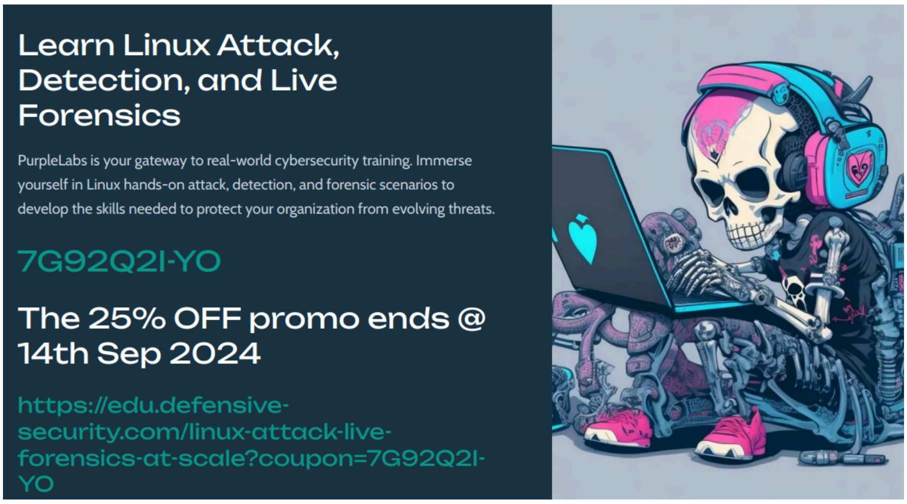
java_debug_wire_proto_load → Monitors for the loading of libjdwp.so

java_libinstrument_load → Triggers when there is an attempt to load libinstrument.so through memory mapping (mmap)

machine_fingerprint → Triggered by unauthorized access to a specific system directories and files that are commonly used to gather information about the underlying machine hardware and network configuration, ex. /sys/class/dmi/id, /sys/class/net, /proc/ioports, etc.

os_fingerprint → Identifies attempts to gather detailed information about the operating system on which it is running

os_status_fingerprint → Identifies attempts to gather detailed information about the operating system's status, which can be indicative of reconnaissance activities within a compromised system



package_repo_config_modification → Triggered when there are modifications to critical package management configuration files across various Linux distributions. This includes files like /etc/apt/sources.list, /etc/yum.conf, and others

pam_config_modification → Identifies unauthorized modification attempts that have been made on critical PAM configuration files located in /etc/pam.d/ and /lib/security/

sched_debug_access → Detects an attempt that was made to access the /proc/sched_debug file on a Linux system

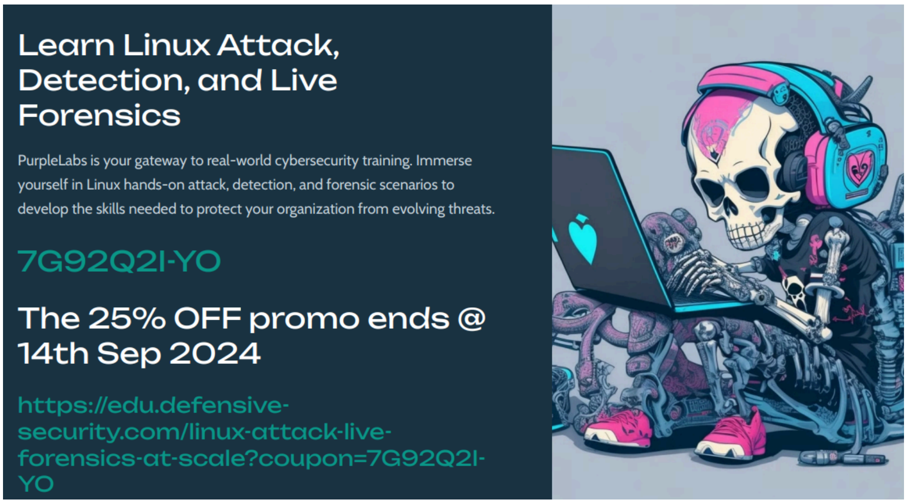
shell_config_modification → Identifies unauthorized or suspicious modifications to critical shell configuration files across various user and system profiles, ex. .bashrc, .profile, and /etc/profile

ssl_certificate_access → Detects unauthorized or unusual access to SSL certificate files. ex. /etc/ssl/, /etc/ca-certificates/, /usr/share/ca-certificates/, /usr/local/share/ca-certificates/

sudoers_modification → Identifies modifications to sudoers configuration

sysrq_access → Triggered when there is an access to /proc/sys/kernel/sysrq or /proc/sysrq-trigger

unprivileged_bpf_config_access → triggered when there is an attempt to access BPF configuration files without the appropriate privileges.



Velociraptor IR → a tool for collecting host-based state information using The Velociraptor Query Language (VQL) queries:

Linux.Detection.MemFD → looks for processes that have been executed from memory via memfd_create()

Linux.Detection.Yara.Process → Runs Yara over processes in memory

Generic.Detection.Yara.Glob → Returns a list of target files then runs Yara over the target list

Generic.Detection.Yara.Zip → Runs Yara on embeded compressed files

Linux.Proc.Modules → Lists loaded kernel modules via /proc/modules

Linux.Sys.Maps → Parses the /proc/PID/maps to emit all mapped files into the process

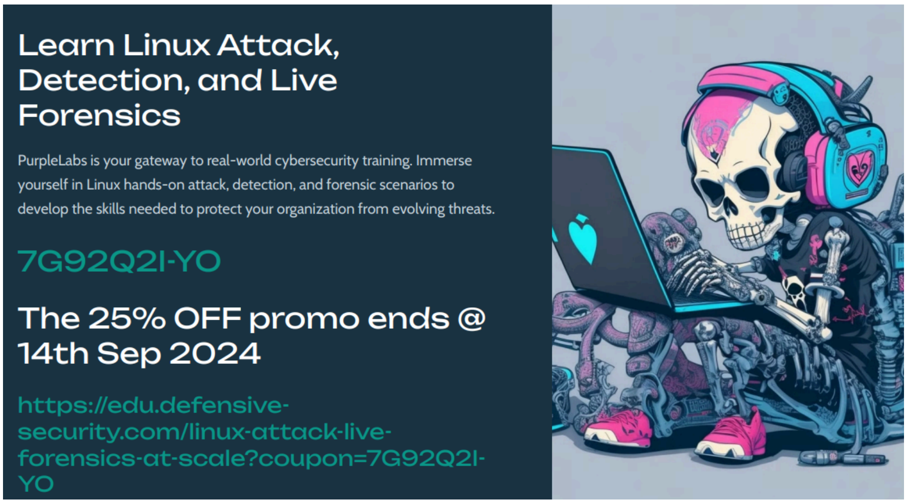
Linux.Sys.Pslist → List processes and their running binaries.

Linux.Sys.SUID → Searches for files with setuid or setgid flag

Generic.Detection.WebShells → Looks for evidence of a web shell being present on the system (based on Yara rules)

Linux.Memory.AVML → Acquires a full memory image in LiME output format.

Linux.Detection.IncorrectPermissions → Checks a number of files and directories to verify whether they have the expected owner, group owner and mode



Linux.Network.NM.Connections → Lists the NetworkManager state, all configured connections and their settings

Linux.Debian.GPGKeys → Extract keys, fingerprints and identities from GPG keys.

Linux.Debian.AptSources → Searches for all apt sources file

Linux.Debian.Packages → Parses dpkg status file.

Linux.RHEL.Packages → Parses packages installed from dnf/yum/rpm

Generic.Forensic.LocalHashes.Query

Generic.Forensic.LocalHashes.Init

Generic.Forensic.LocalHashes.Glob → maintains a local database of file hashes. It is then possible to query this database using the Generic.Forensic.LocalHashes.Query

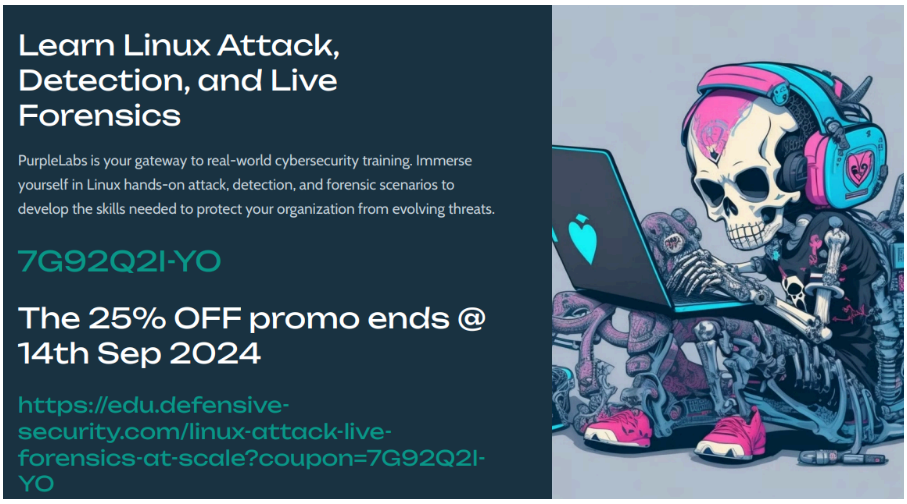
Linux.PrivilegeEscalationDetection → identifies processes running as root that were spawned by processes not running as root

Exchange.Linux.Kunai → Parses the Kunai log file

Linux.LogAnalysis.ChopChopGo → Leverages ChopChopGo to enable usage of Sigma rules to facilitate detection within Linux logs

Generic.Collection.UAC → Leverages UAC (Unix-like Artifacts Collector) to collect artifacts from Unix-like systems

Linux.Collection.Autoruns → Collects various autorun files based on TriagePersistence



Linux.Collection.BrowserExtensions → Collects Browser Extensions based on TriageWebBrowserExtensions

Linux.Collection.BrowserHistory → Collects Browser History based on TriageWebBrowserHistory

Linux.Collection.DBConfig → Collects database configurations based on TriageDatabaseConfigsAndLogs

Linux.Collection.History→ Collects history files from unix/linux utilities based on TriageHistory

Linux.Collection.NetworkConfig → Collects network config files based on TriageNetwork

Linux.Collection.SysConfig → Collects system configurations based on TriageSystemConfiguration

Linux.Collection.SysLogs → Collects system logs based on TriageSystemLogs

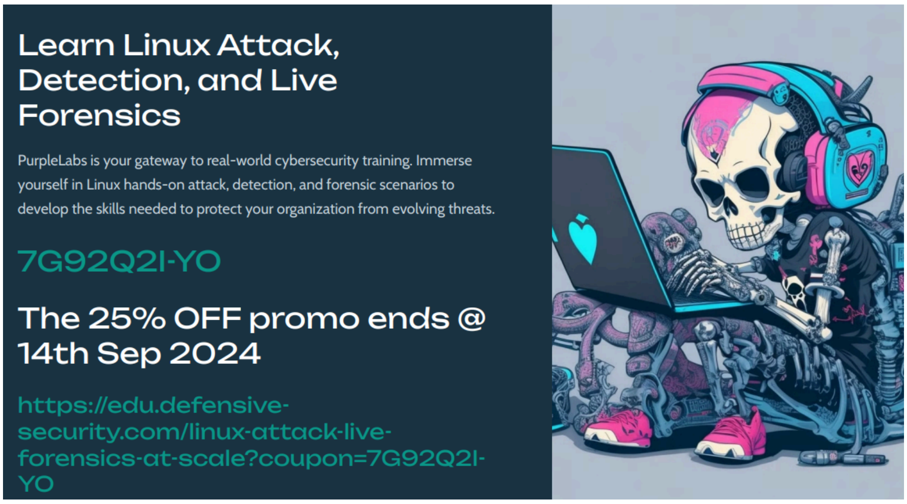
Linux.Collection.UserConfig → Collects user configurations and based on TriageUserConfiguration

Linux.System.BashLogout → Captures Bash logout files for examination of abnormal activity

Linux.Sys.BashShell → Allows running arbitrary commands through the system shell

Linux.Sys.LastUserLogin → Finds and parses system wtmp files

Linux.Sys.Crontab → Displays parsed information from crontab



Linux.Forensics.RecentlyUsed → Parses the 'recently-used.xbel' XML file for all standard Linux users

Linux.Sys.APTHistory → Checks the log of software installation/removal/upgrades

Linux.Sys.JournalCtl → Parses the output of the journalctl command

Linux.Forensics.Journal → Parses the binary journal logs

Linux.Sys.SystemdTimer → Lists and parses content of Systemd timers

Linux.Remediation.Quarantine → Quarantines a Linux host using iptables rules

Linux.Detection.ConfluenceLogs → Enables grep of Linux logs and targets strings observed in exploitation of CVE-2022-26134

Linux.Detection.CVE20214034 → Lists processes running as root that were spawned by processes that are not running as root

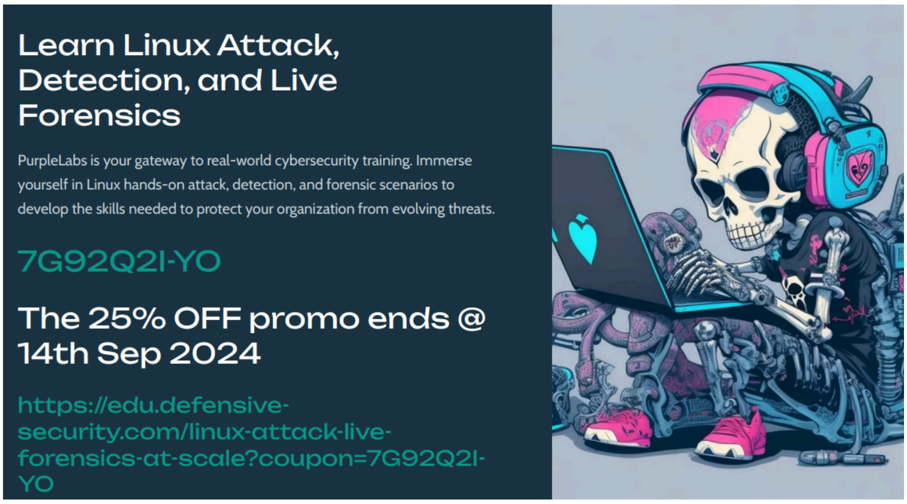
Linux.Sys.LogHunter → Enables grep of Linux, MacOS, and Windows logs. Parameters include SearchRegex and WhitelistRegex as regex terms

Linux.Sys.Services → Parses services from systemctl

Linux.Sys.Users → Gets user-specific information like homedir, group etc from /etc/passwd

Linux.Users.InteractiveUsers → Gets the interactive users from a Linux host

Linux.Users.RootUsers → Detects users added to the sudo group



Linux.Sysinternals.SysmonEvent → Parses syslog for Sysmon events on Linux using a unix domain socket

Linux.Sysinternals.Sysmon → Parses syslog for Sysmon events on Linux

Generic.Detection.log4jRCE → Detects the exploitation attempts against log4j RCE vulnerability CVE-2021-44228

Linux.Collection.CatScale → Leverages Cat-Scale to collect many different artifacts from a Linux host

Linux.Applications.WgetHSTS → Gets a wget HSTS log file in a user's home directory

Linux.Network.Netstat → Parses /proc and reveal information about current network connections

Linux.Network.NetstatEnriched → Reports network connections, and enriches with process information

Linux.Network.PacketCapture → Leverages tcpdump to natively capture packets

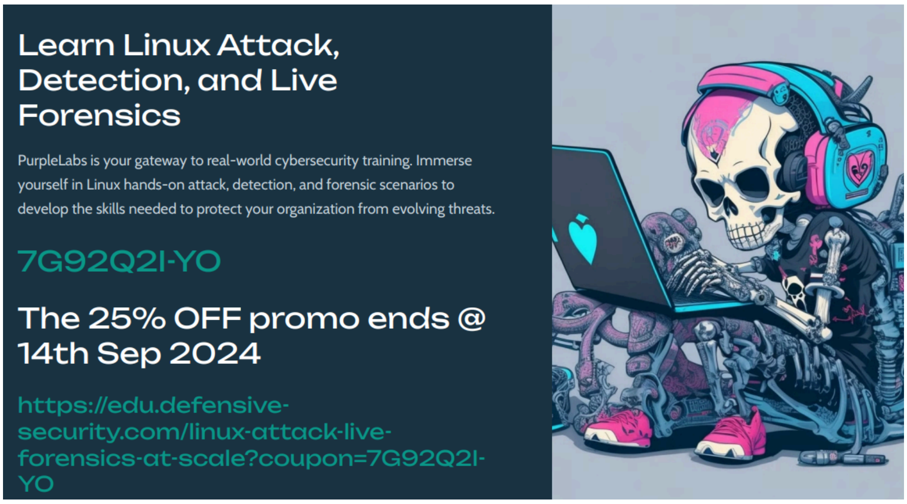
Linux.OSQuery.Generic → Executes OSquery query

Generic.System.Pstree → Displays the call chain for every process on the system by traversing the process's parent ID

Linux.Memory.Acquisition → Acquires a full memory image by LiiME

Linux.Triage.ProcessMemory → Dumps process memory and upload to the server

Linux.Volatility.Create.Profile → Creates Volatility Framework profile to the Debian / Ubuntu OS



Exchange.Linux.Detection.BPF → Parses /proc/*/fd files and looks for processes with anon_inode:bpf-map

Exchange.Linux.System.PAM → Enumerates applicable lines from the files that reside in /etc/pam.d/

Linux.Applications.Docker.Info → Gets Dockers info by connecting to its socket.

Linux.Applications.Docker.Version → Get Dockers version by connecting to its socket

Linux.Detection.AnomalousFiles → Detects anomalous files in a Linux filesystem (hidden, large, SUID)

Linux.Mounts → Lists mounted filesystems by reading /proc/mounts

Linux.Proc.Arp → Lists ARP table via /proc/net/arp

Linux.Search.FileFinder → Finds files on the filesystem using the filename or content

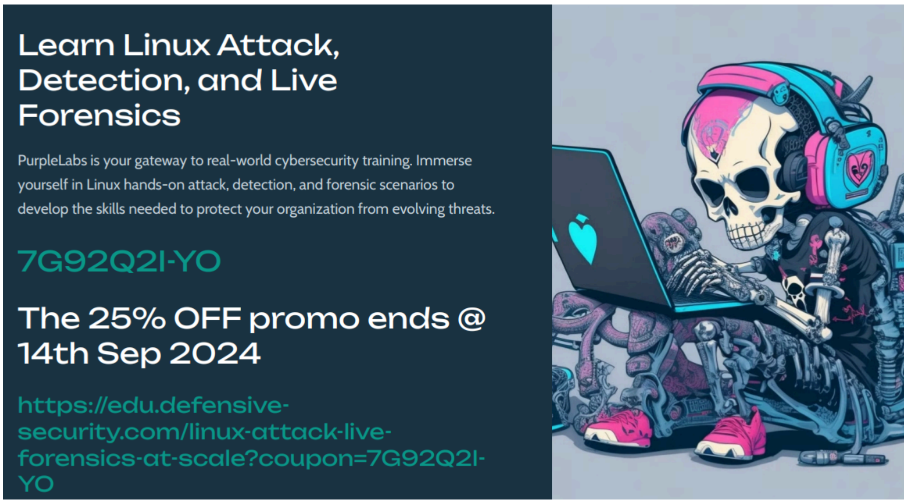
Linux.Ssh.AuthorizedKeys → Finds and parses ssh authorized keys files

Linux.Ssh.KnownHosts → Finds and parses ssh known hosts files

Linux.Ssh.PrivateKeys → Searches for private keys in the usual locations and also records if they are encrypted or not

Linux.Syslog.SSHLogin → Parses the auth logs to determine all SSH login attempts

Linux.Detection.SSHKeyFileCmd → Parses ~/.ssh/authorized_keys and ~/.ssh/id*.pub looking for the command option

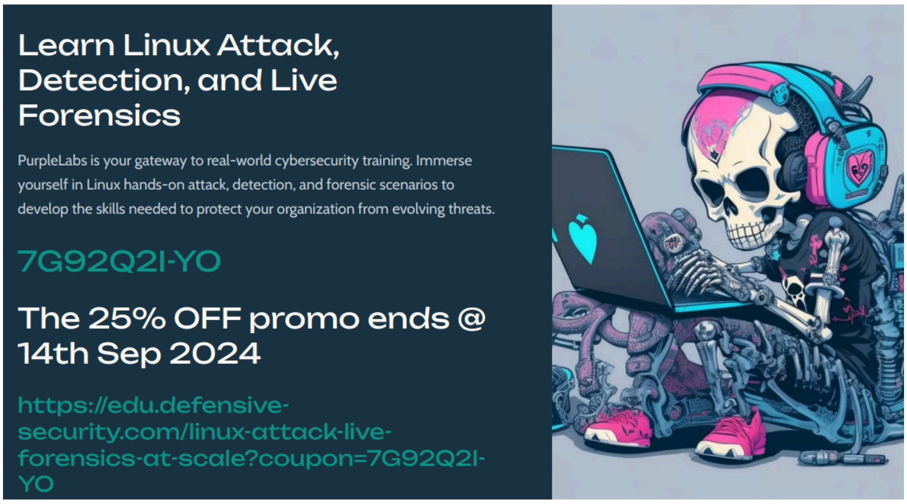


to detect potential persistence

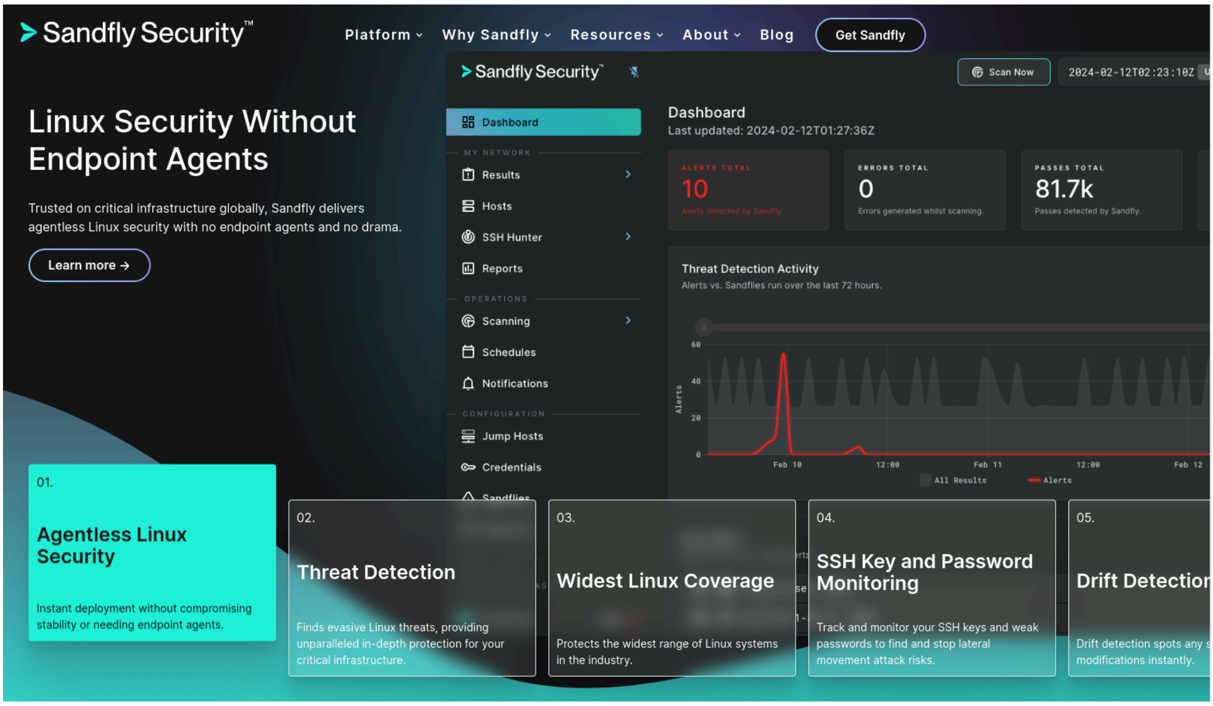
Linux.ExtractKthread → Parses `/proc/[0-9]*/status` files and extracts the ProcessName and Kthread values.

Linux.Forensics.EnvironmentVariables.v3 → Detects potential persistence mechanisms on Linux systems by analyzing environment variable files and login scripts

Linux.Network.Nethogs→ Lists all processes that produce (non-local) network traffic on the client, leveraging the Nethogs process tracker



SANDFLY SECURITY → Sandfly is an agentless, instantly deployable, and safe Linux security monitoring platform. Sandfly automates security investigation and forensic evidence collection on Linux. To learn more visit: https://sandflysecurity.com/. Top 30 Sandfly modules out of thousands:



Linux Security Without Endpoint Agents

Trusted on critical infrastructure globally, Sandfly delivers agentless Linux security with no endpoint agents and no drama.

Learn more →

Dashboard

Last updated: 2024-02-12T01:27:36Z

ALERTS TOTAL 10 Alerts detected by Sandfly.

ERRORS TOTAL 0 Errors generated whilst scanning.

PASSES TOTAL 81.7k Passes detected by Sandfly.

01. Agentless Linux Security

Instant deployment without compromising stability or needing endpoint agents.

02. Threat Detection

Finds evasive Linux threats, providing unparalleled in-depth protection for your critical infrastructure.

03. Widest Linux Coverage

Protects the widest range of Linux systems in the industry.

04. SSH Key and Password Monitoring

Track and monitor your SSH keys and weak passwords to find and stop lateral movement attack risks.

05. Drift Detection

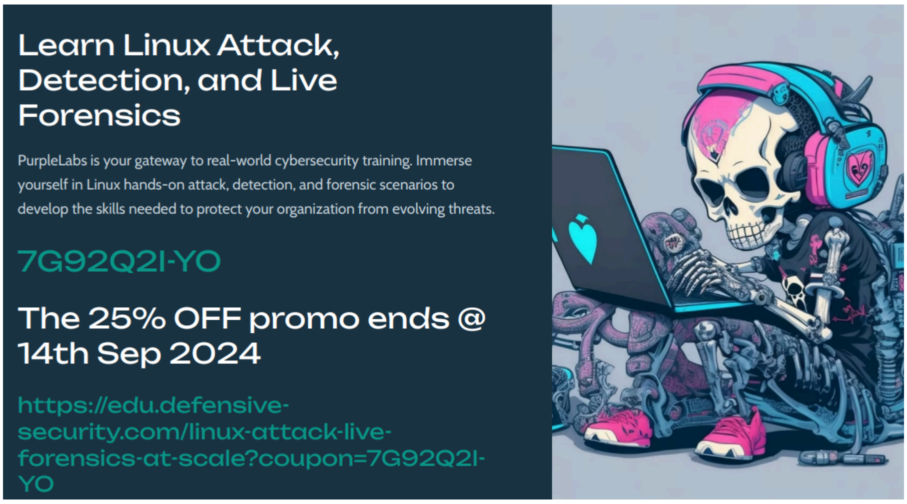
Drift detection spots any s modifications instantly.

---

process_deleted → Looks for processes that are running, but the executable has been deleted from the disk

---

process_environ_history_anti_forensics → Looks for processes with environment variables indicating anti-forensics are being used to conceal command history



Learn Linux Attack, Detection, and Live Forensics

PurpleLabs is your gateway to real-world cybersecurity training. Immerse yourself in Linux hands-on attack, detection, and forensic scenarios to develop the skills needed to protect your organization from evolving threats.

7G92Q2I-YO

The 25% OFF promo ends @ 14th Sep 2024

https://edu.defensive-security.com/linux-attack-live-forensics-at-scale?coupon=7G92Q2I-YO

https://edu.defensive-security.com/linux-attack-live-forensics-at-scale?coupon=7G92Q2I-YO

process_running_from_tmp_dir → Looks for processes that are running out of the system temp directories

process_running_from_dev_dir → Looks for processes that are running out of the system /dev directories

process_running_from_hidden_dir_anywhere → Looks for processes that are running out of a hidden directory anywhere on the system

process_running_from_suspicious_path → Looks for processes with environment variables indicating anti-forensics are being used to conceal command history

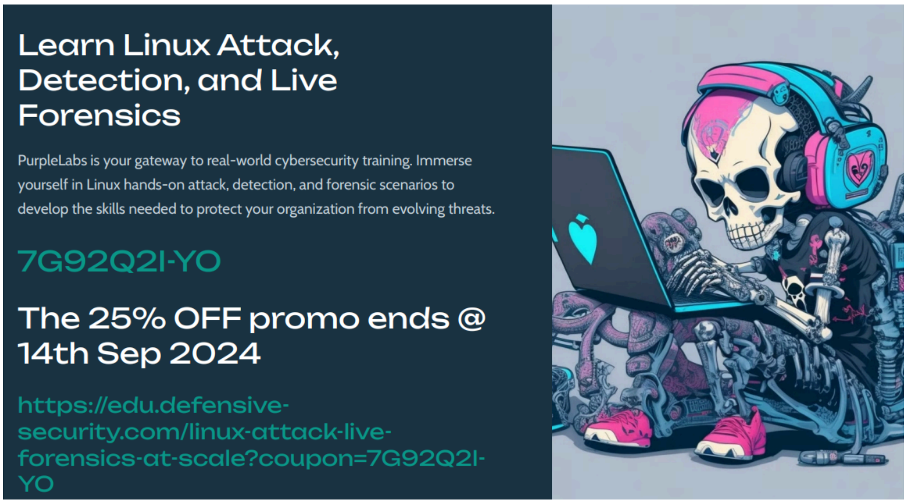
process_running_from_root_homedir_dir → Looks for processes that are running out of the /root directory

process_running_from_system_dir → Looks for processes that are running out of /boot, /sys and /lost+found directories

process_running_hidden_name → Looks for processes that are named as a Unix hidden file that are running (e.g. period as the start of name)

process_masquerade_extension_suspicious → Looks for processes that are running with an extension of their name that normally wouldn't be on a system binary.

process_name_suspicious → Looks for processes that are running with a suspicious name to hide the binary on the disk



process_entropy_high → Looks for processes with high entropy indicating it may be packed or encrypted which is common with malware and malicious activity.

process_binary_immutable → Looks for any process with a binary that is marked as immutable

process_masquerade_kernel_thread_* → Looks for processes hiding with a name to appear to be a kernel thread ([brackets])

process_running_hidden_stealth → Looks for processes that have been hidden by a stealth rootkit

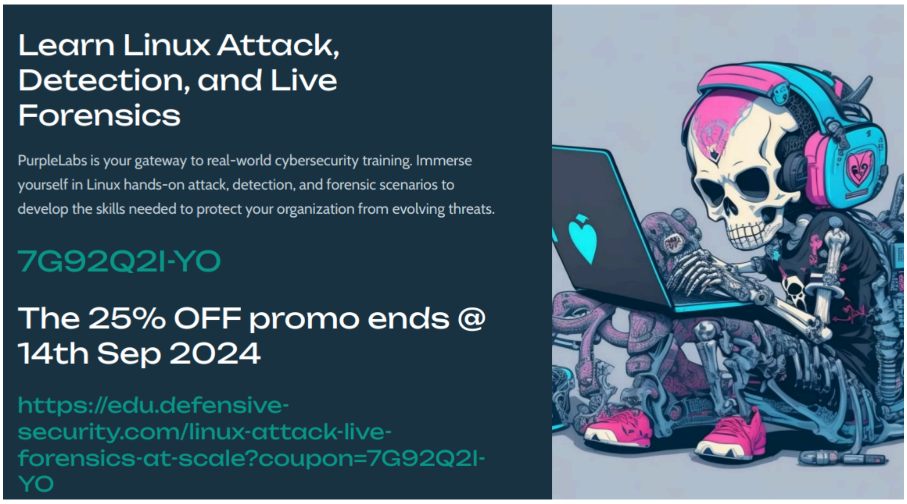
user_ssh_authorized_keys_immutable → Looks for users that have an SSH authorized_keys file that is set as immutable

policy_user_ssh_authorized_keys_duplicates_found → Looks for users that have SSH authorized_keys key data that are duplicates

file_binary_in_tmp_dir → Looks for executable files in the top-level system temp directories (no recursion)

file_hidden_bin → Looks for any kind of hidden file under system binary directories which is unusual behavior

systemd_exec_from_hidden_dir_anywhere → Looks for systemd units that run commands in a hidden directory anywhere on the system



user_default_user_ssh_authorized_keys_present → Looks for default Linux system users that have a SSH authorized_keys file presents that could allow login

file_binary_entropy_high_in_dev_dir → Looks for high entropy packed or encrypted executable files in system /dev directories

kernel_module_hidden → Kernel modules that appear to be trying to hide themselves

user_password_auditor_password_is_username → Looks for users with a password that is the same as their username

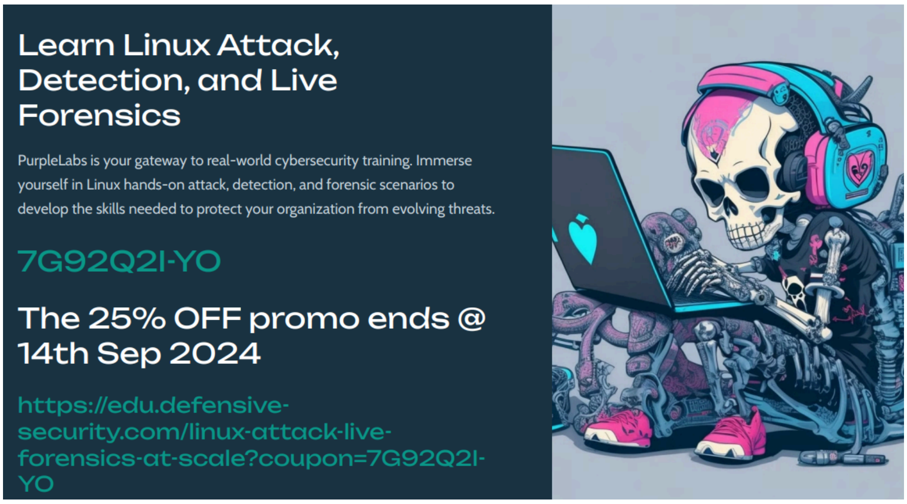
user_default_user_password_present → Looks for default Linux system users that have a password hash present that could allow login

policy_user_password_auditor_top_worst_small_list → Looks for users with a password that is one of the top worst passwords (~100 word list)

process_persistence_cron_malicious → Looks for cron tasks that are suspicious or malicious

kernel_module_file_missing → Kernel modules that are loaded but don't have .ko files in /lib/modules/

dirs_hidden_dev_shm → Looks for hidden directories in /dev/shm



process_shell_running_empty_file_descriptors_command_mode → Looks for processes running with empty file descriptors

process_environ_proc_home_dir → Searches for suspicious home directory location in process environment

systemd_exec_args_base64 → Looks for systemd units that contain base64 encoded data to obfuscate entries

systemd_exec_args_obfuscation → Looks for systemd units that are using commands that obfuscate data

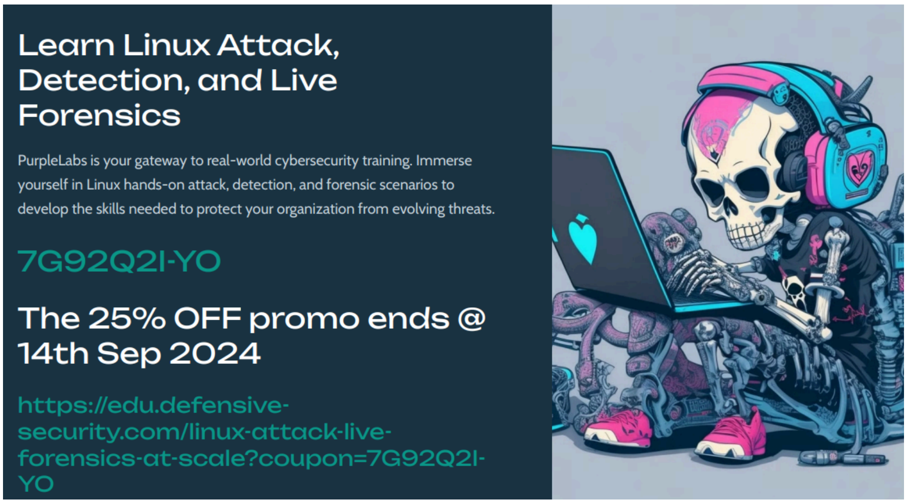
systemd_exec_args_malicious → Looks for systemd units that have indications of suspicious or malicious use

systemd_exec_args_shell_execution → Looks for systemd units that executes another shell via the command (-c) mode

process_shell_running_kthread_spawned_command_mode → Looks for shell processes in command (-c) mode started by the kthread process

policy_user_ssh_private_key_in_user_home_dir → Searches for SSH private keys in any user's SSH directory

policy_cpu_load15_high → Finds overloaded systems or systems with suspiciously high CPU activity



DFIR/Triage Tools:

UAC → Live Response collection script for Incident Response

LinuxCatScale → Incident Response collection and processing scripts with automated reporting scripts

Fennec → Artifact collection tool for *nix systems

varc → Volatile Artifact Collector collects a snapshot of volatile data from a system

chkrootkit → Checks for signs of a rootkit

rkhunter → Rkhunter Malware Scanner for linux

lynis → Security auditing tool for Linux, macOS, and UNIX-based systems

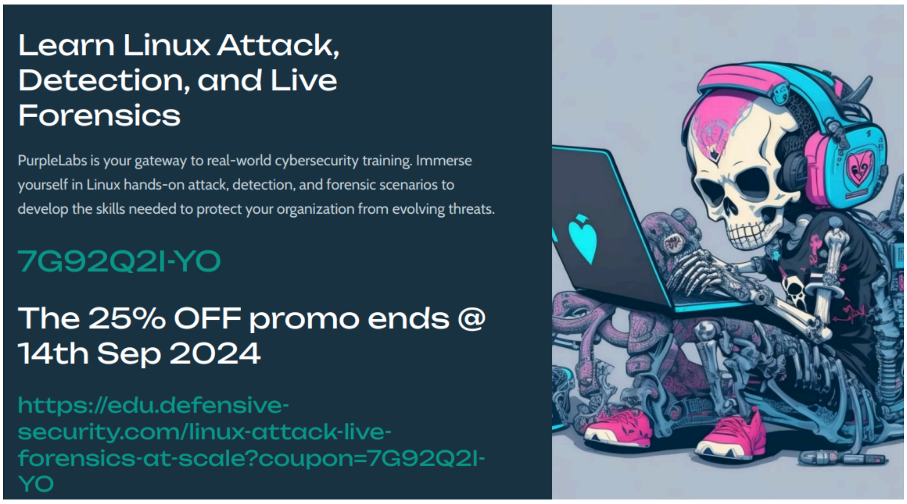
Unhide → Forensic tool to find hidden processes and TCP/UDP ports by rootkits

GRR Rapid Response → Incident response framework focused on remote live forensics

sandfly-file-decloak → Decloak Linux stealth rootkits hiding data with this simple memory mapped IO investigation tool

sandfly-process-decloak → Utility to quickly scan for Linux Process IDs (PIDs) that are hidden by common and not-so-common loadable kernel module stealth rootkits and decloak them so they are visible

sandfly-entropyscan → Entropy scanner for Linux to detect packed or encrypted binaries related to malware



LINKS:

- https://github.com/falcosecurity/falco
- https://github.com/aquasecurity/tracee
- https://github.com/cilium/tetragon
- https://listendev.github.io/argus/dev/overview/
- https://github.com/Sysinternals/SysmonForLinux/
- https://why.kunai.rocks/
- https://github.com/chainguard-dev/osquery-defense-kit
- https://github.com/tstromberg/sunlight
- https://github.com/Velocidex/velociraptor
- https://github.com/lkrg-org/lkrg
- https://github.com/sandflysecurity/sandfly-file-decloak
- https://github.com/sandflysecurity/sandfly-processdecloak
- https://github.com/tclahr/uac