DEFENSIVE SECURITY

Proaktywna analiza metod persystencyjnych w systemie Linux.

Leszek Miś
lm@defensive-security.com

whoami

- Security Researcher / Founder / uid=0 @ Defensive Security
- Offensive Security Certified Professional (OSCP)
- Red Hat Certified Architect/RHCSS/RHCX/Sec+
- Trener / wykładowca podczas Black Hat US, Hack In The Box AMS / Singapore / Abu Dhabi, OWASP Appsec US, Flocon US, BruCON, Confidence PL
- Obszary zainteresowania:
  - PurpleLabs Cyber Range Playground
  - Adversary Emulations and Post-Exploitation Red/Blue Actions
  - Threat Hunting and Incident Response
  - Behavioral / Statistic / ML network analysis → Features Extraction
  - Hardening of Linux / Web Application / Infrastructure
  - Penetration testing / OSINT / Security audits
  - Open Source Security Software

Agenda

I. Wprowadzenie:

- Czym jest persystencja?
- Przegląd lokalizacji persystencji w systemie Linux w ujęciu Purple Team
- Zrozumienie kontekstu

II. Detekcja, telemetria oraz analityka w bezpośrednim odniesieniu do:

- FIM → Monitoring integralności plików
- auditd
- EBPF → Sysmon / Tracee / Falco / Sysdig
- OSQuery
- Velociraptor
- Sandfly
- CLI

III. Proaktywne skanowanie DFIR

I. Czym jest persystencja?

Persystencja 101

- Persystencja to nic innego jak “trwały dostęp”, sposób na pozostanie w sieci / w obrębie hosta

Home > Tactics > Enterprise > Persistence

Persistence

The adversary is trying to maintain their foothold.

Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code.

- https://attack.mitre.org/tactics/TA0003/

Persystencja 101

- Dlaczego persystencja jest istotna z punktu widzenia atakującego?
    - Brak potrzeby ponownego uzyskiwania dostępu od zera:
        - Nowy maldoc phishing / drive by download / nowa strategia / nowe C2
        - Exploitowany CVE mógł zostać załatany
        - Hardening → np. ograniczenia dla LOLBAS / lolbins / sudo
        - Hasła i klucze SSH mogły zostać zmienione
        - Lepsze alertowania NIDS / NIPS pod kątem wysyłanych payloadów, ataków typu BF / Password spraying
    - Możliwość ukrycia się i odczekania na dobry moment, np. celem eksfiltracji, szyfrowania danych, kopania Monero lub uzyskania uprawnień root:
        - → CVE-2021-4034 →https://haxx.in/files/blasty-vs-pkexec.c
    - Stabilny dostęp na wielu różnych warstwach / protokołach

Persystencja 101

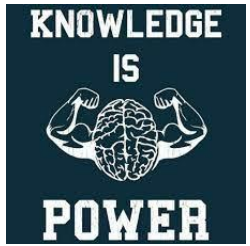
- Persystencja często pomijana jest w testach penetracyjnych, ale tym samym jest integralnym elementem emulacji atakujących / testów opartych na BAS (Breach and Attack Simulations)

Persystencja 101

- “Obrońcy muszą znać tysiące sposobów, na które system może zostać skompromitowany. Atakujący muszą znać tylko ten jeden właściwy.”

- “Atakujący muszą znać tysiące sposobów, aby zatrzeć po sobie ślady. Obrońcy muszą zauważyć tylko jedną nieprawidłowość.”

- Z punktu widzenia defensywnego można do tematu podejść inaczej:
  - **Persystencja to kolejna okazja do wykrycia!**

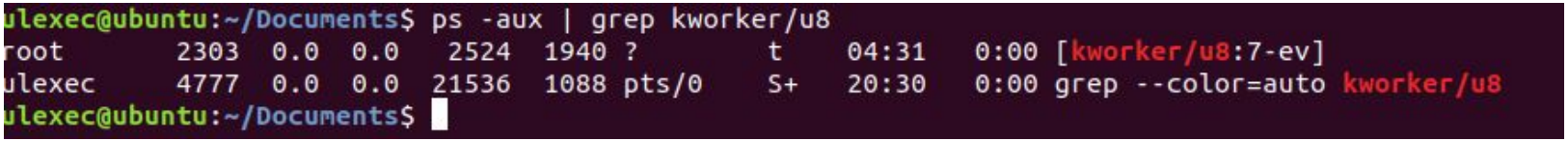


Lokalizacje persystencji

- Wysokopoziomowo:
  - Pliki
  - Katalogi
  - Procesy
  - Użytkownicy
  - Logi

Lokalizacje persystencji

- Podejrzane procesy:
  - Proces nadal działa, ale binarka została usunięta
  - Proces komunikuje się poprzez sieć
  - Wysokie obciążenie CPU
  - Niespotykana nazwa procesu
  - Proces o nazwie bardzo podobnej lub identycznej do procesów systemowych czy wątków kernelowych np. **[kworker/1:2H]**
  - Poprawny proces systemowy, ale ze wstrzykniętym kodem poprzez:
    - ptrace() → https://github.com/gaffe23/linux-inject
    - dlinject.py → https://github.com/DavidBuchanan314/dlinject



Lokalizacje persystencji

- Podejrzane katalogi:
  - Ukrywanie narzędzi atakujących, złośliwych kodów, skryptów
  - Ukrywanie wykradzionych danych przed procesem eksfiltracji
  - Staging do następnych etapów ataku, np. pozostawienie backd00r_socket.s
- Wykorzystywane lokalizacje:
  - /dev/, /dev/shm, /bin, /sbin, /usr/lib/*, /etc/, /lib64/*, /usr/lib/*
  - /home, ~/public_html, /tmp/, /var/tmp, /var/log, /var/spool
- Podejrzane nazwy plików i katalogów:
  - ...
  - ..%
  - spacja
  - spacja kropka,
  - kropka, spacja, kropka
  - kropka, kropka, kropka, /

Lokalizacje persystencji

- Skrypty startowe:
    - /etc/rc.local
    - /etc/init.d/
    - system:
        - /etc/systemd/system/*
        - /lib/systemd/system/*
    - user:
        - /etc/systemd/user/*
        - /lib/systemd/user/*

```
# find / -path "*/systemd/system/*.service" -exec grep -H -E "ExecStart|ExecStop|ExecReload" {} \;
2>/dev/null

# find / -path "*/systemd/user/*.service" -exec grep -H -E "ExecStart|ExecStop|ExecReload" {} \;
2>/dev/null
```

Lokalizacje persystencji

- cron scheduler:
  - /etc/crontab
  - /etc/cron.d/*
  - /var/spool/cron/*
  - /etc/cron.daily/*
  - /etc/cron.hourly/*
  - /etc/cron.monthly/*
  - /etc/cron.weekly/*
  - @reboot

Lokalizacje persystencji

- Podejrzane flagi / bity na plikach/katalogach:
  - chattr / lsattr (immutable bit):
    - # lsattr /usr/bin/backdoor

      ----i---------- /usr/bin/backdoor

- sockety z uprawnieniami uid=0:
  - server = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
  - server.bind("**/dev/shm/.purplelabs.s**")

Lokalizacje persystencji

- Podmienione / zmodyfikowane pliki systemowe:
  - weryfikacja integralności plików pochodzących od pakietów:
    - RPM:
      - rpm -Va
    - DEB:
      - dpkg --verify

Lokalizacje persystencji

- Konfiguracja powłoki:
  - https://github.com/SigmaHQ/sigma/blob/master/rules/linux/auditd/lnx_auditd_alter_bash_profile.yml

File	Description
/etc/profile	Systemwide files executed at the start of login shells
/etc/profile.d/	All .sh files are executed at the start of login shells
/etc/bash.bashrc	Systemwide files executed at the start of interactive shells
/etc/bash.bash_logout	Systemwide executed as a login shell exits
~/.bashrc	User-specific startup script executed at the start of interactive shells
~/.bash_profile, ~/.bash_login, ~/.profile	User-specific startup script, but only the first file found is executed
~/.bash_logout	User-specific clean up script at the end of the session

Lokalizacje persystencji

- User mode rootkit → głównie poprzez LD_PRELOAD:
  - ukrywanie plików, procesów, połączeń sieciowych, zalogowanych użytkowników
  - Przykłady:
    - vlany
    - beurk
    - Jynx2
    - azazel
    - Umbreon-Rootkit

- Podmiana dynamicznego linkera / linker preloading:
  - /lib64/ld-linux.so -> evild_ld:
    - https://tmpout.sh/2/6.html

https://github.com/milabs/awesome-linux-rootkits#see_no_evil-user-mode-rootkits

Lokalizacje persystencji

- User mode rootkit:
    - PAM-based + Telegram Exfil
        - /etc/pam.d/ , /lib64/security/
        - https://github.com/mthbernardes/sshLooterC/blob/master/looter.c
    - SSHD backdoor:
        - https://blog.xpnsec.com/linux-process-injection-aka-injecting-into-sshd-for-fun/
    - HTTPD:
        - mod_backdoor HTTPD Server:
            - https://github.com/VladRico/apache2_BackdoorMod
        - mod_authg:
            - https://github.com/ChristianPapathanasiou/apache-rootkit

Lokalizacje persystencji

- Webshells:
  - uruchamianie kodu z kontekstu aplikacji / serwera HTTP:
    - SOCKS proxy z poziomu PHP/ASP/JSP:
      - https://github.com/sensepost/Regeorg
  - Warto zwrócić uwagę na → slopShell:
      - https://github.com/oldkingcone/slopShell
  - Generyczna detekcja webshelli na bazie relacji parent-child dla procesów:
    - https://github.com/SigmaHQ/sigma/blob/master/rules/linux/process_creation/proc_creation_lnx_webshell_detection.yml

Lokalizacje persystencji

- Kernel mode rootkit → LKM → Loadable Kernel Modules:
  - ukrywanie plików, procesów, połączeń sieciowych
  - ukrywanie modułu /proc/modules
  - aktywacja keyloggera / drop2shell poprzez wysłanie specjalnego sygnału / pakietu sieciowego
  - Przykłady:
    - suterusu
    - rkduck
    - reptile
    - Diamorphine
    - krf
    - Umbra
    - Drovorub

https://github.com/milabs/awesome-linux-rootkits#hear_no_evil-kernel-mode-rootkits

Lokalizacje persystencji

- Użytkownicy:
  - /etc/passwd , /etc/shadow, /etc/group, /etc/gshadow
  - Konta z ustawionym hasłem:
    - SELECT password_status, username, last_change

      FROM shadow

      WHERE password_status = 'active';
  - Konta z ustawionym uid=0
  - Konta z ustawionym /bin/sh, /bin/bash, /bin/zsh, etc.

- Klucze SSH:
  - /home/*/.ssh/authorized_keys
  - /root/.ssh/authorized_keys

Lokalizacje persystencji

- Ciekawszą drogą uzyskania szerszej, centralnej persystencji do infrastruktury linuksowej jest infekcja stacji roboczej administratora → np Windows 10/11 →
    - Outlook C2:
        - email jako wyzwalacz do uruchomienia kodu:
            - wykorzystuje Microsoft.Office.Interop.Outlook namespace
            - Pełny dostęp do struktury i danych Outlooka
                - https://github.com/S4R1N/BadOutlook

- Pozostają jeszcze ataki typu:
    - Modyfikacja initrd/initramfs
    - Low Level Attacks/Firmware/BIOS/UEFI

Rekomendowany kierunek

- Sigma Rules:
  - Generyczny format opisu zdarzeń bazujący na logach
  - Konwenter reguł Sigma do dowolnego silnika SIEM:
    - sigmac
    - https://uncoder.io/
  - Sigma for Linux:
    - https://github.com/SigmaHQ/sigma/tree/master/rules/linux

Detekcja

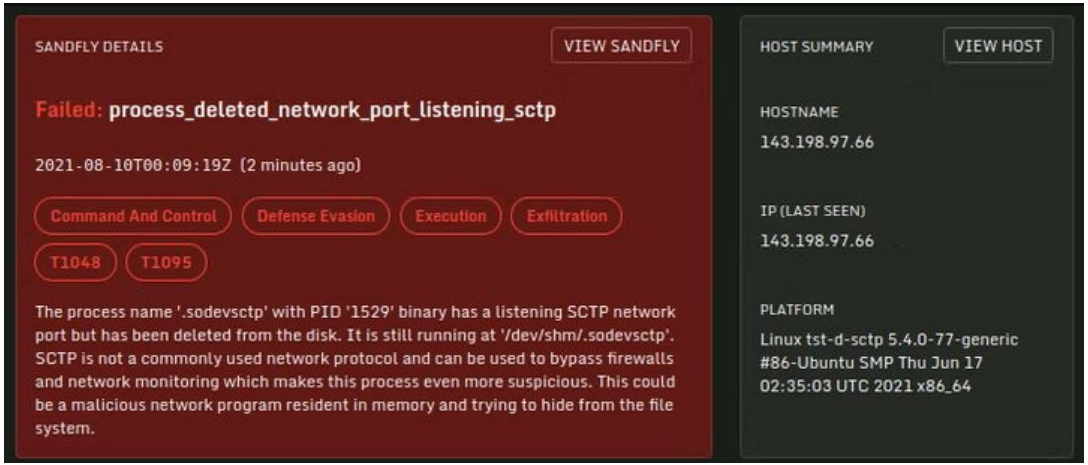
- eBPF:
  - Falco
  - Tracee
  - Sysmon for Linux
- Wazuh
- auditd
- OSQuery
- Velociraptor:
  - Linux.Collection.CatScale
  - Yara na plikach / na procesach
- Strelka
- unhide

Linki

- VMware Exposing Malware in Linux-Based Multi-Cloud Environments:
  - https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/docs/vmw-exposing-malware-in-linux-based-multi-cloud-environments.pdf
- My Methods To Achieve Persistence In Linux Systems:
  - https://flaviu.io/advanced-persistent-threat/
- Advanced Persistent Threat Techniques Used in Container Attacks:
  - https://blog.aquasec.com/advanced-persistent-threat-techniques-container-attacks
- Linux Compromise Detection:
  - https://2018.purplecon.nz/archive/craig-h-rowland/Linux.Compromise.Detection.Presentation.pdf
- Hunting for Persistence in Linux:
  - https://pberba.github.io/security/2021/11/22/linux-threat-hunting-for-persistence-sysmon-auditd-webshell/

Proaktywne skanowanie DFIR

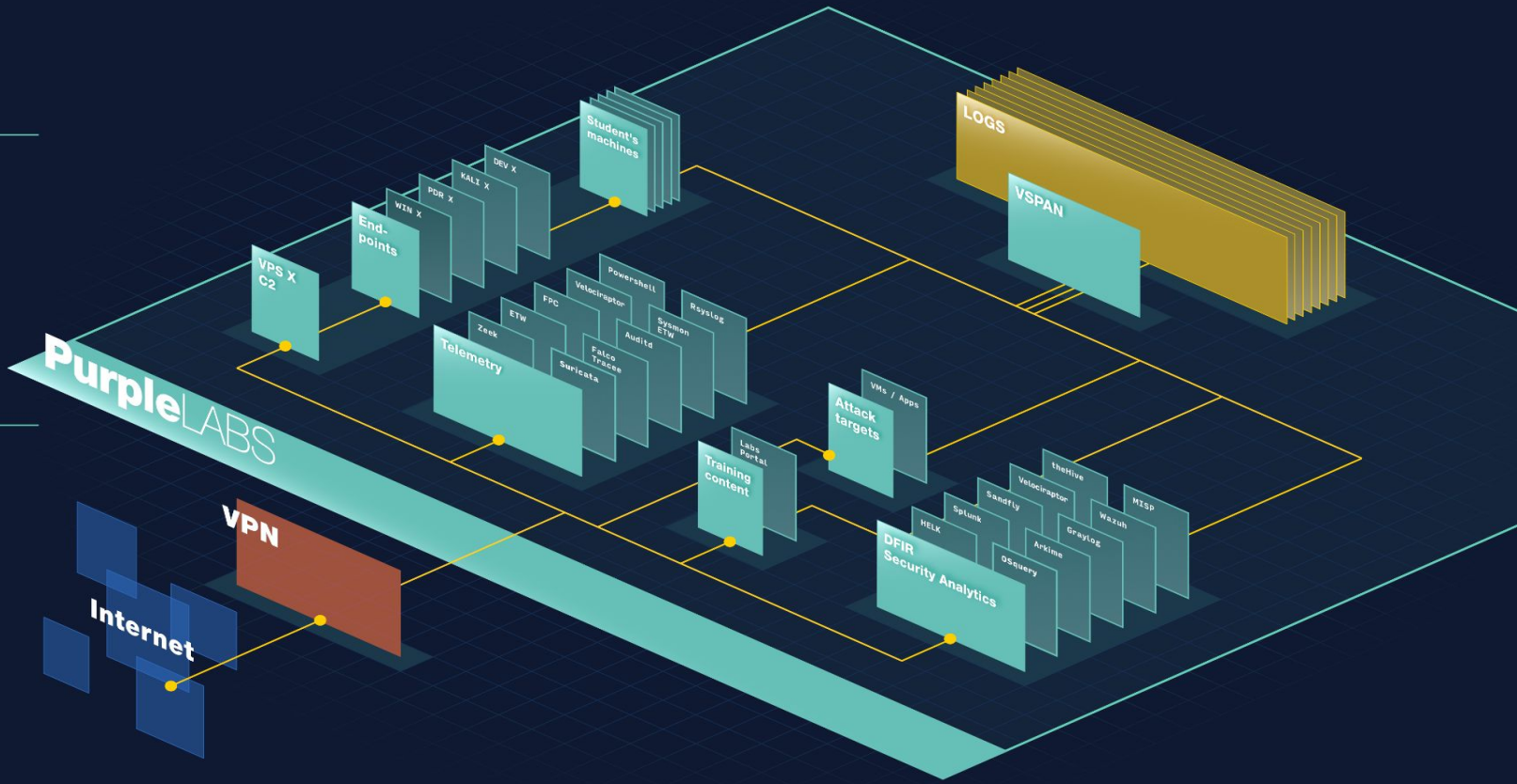
- Sandfly Security + usługa cyklicznego skanowania:
  - skaner wykrywający podejrzane zachowanie dowolnego Linuksa
  - bezagentowy, ponad 1000 definicji wykrywania anomalii
  - kontekstowy opis wyników + mapowanie MITRE Attack Framework
  - **https://www.sandflysecurity.com/**



PurpleLABS

Cyber Range Playground with Hands-On Labs

PurpleLABS is a dedicated virtual infrastructure for conducting detection and analysis of attackers' behavior in terms of used techniques, tactics, procedures, and offensive tools. The environment has been created to serve the constant improvement of competences in the field of threat hunting (threat hunting) and learning about current trends of offensive actions (red teaming) vs direct detection (blue teaming).



DEFENSIVE SECURITY

Dziękuję za uwagę i zapraszam do kontaktu!

Leszek Miś
lm@defensive-security.com
https://defensive-security.com