Falcoルール

https://github.com/falcosecurity/falco/blob/master/rules/falco_rules.yaml

2020年12月10日 時点

本文の内容は、Cloud Native Runtime Security であるFalco (https://github.com/falcosecurity/falco/）において、2020年12月10日時点を元に作成したドキュメントです。

はじめに

このドキュメントは、CNCFでデファクトスタンダードとなっているクラウドネイティブランタイムセキュリティの仕組みである Falcoにおける Falcoルール個々の内容について記述しています。Falcoルール(https://github.com/falcosecurity/falco/blob/master/rules/falco_rules.yaml)は、2020年12月10日現在、44名のコントリビューター（日本人のコントリビュータもいらっしゃいます）の面々によって日々、メンテナンスされています。

Falcoルールには、システムコールをソースとしたルール以外に、Kubernetes Auditをソースとしたルール、アプリケーションに特化したルールがあります。また、Cloud Native Security Hub(https://securityhub.dev/）でもFalcoルールは公開されています。

本ドキュメントは、Falcoルール個々の内容について記述しています。記述している、マクロ、リスト、ルールは、Falcoルールファイル内の順番を踏襲しています。Falcoルールの基本については、https://falco.org/jp/docs/rules/ を参照ください。

falco エンジンとルールのバージョニングの詳細については、xxx を参照してください。現在のところ。 この特定のルールファイルは、エンジンバージョン0と互換性があります。　(例: falco releases <= 0.13.1)なので、このままでは required_engine_versionの行がコメントアウトされているので、古いfalcoのリリースとの互換性を維持してください。このルールファイルに最初の互換性のない変更があった場合、この行のコメントを外し、その時点で使用されているfalcoエンジンのバージョンに設定します。

```
- required_engine_version: 7
```

共通マクロ、リスト - 1

Falcoルール（https://github.com/falcosecurity/falco/blob/master/rules/falco_rules.yaml）では、初めにFalcoルール全体で共通して利用するマクロとリストが定義されています。

現在、read/writeはsyscallで無視するため無効になっています。

```
# - macro: write
#   condition: (syscall.type=write and fd.type in (file, directory))
# - macro: read
#   condition: (syscall.type=read and evt.dir=> and fd.type in (file, directory))
```

ほぼ同様の open_write/open_read は、ファイルが読み書きのために開かれているかどうかをチェックします。

```
- macro: open_write
  condition: (evt.type=open or evt.type=openat) and evt.is_open_write=true and fd.typechar='f' and fd.num>=0

- macro: open_read
  condition: (evt.type=open or evt.type=openat) and evt.is_open_read=true and fd.typechar='f' and fd.num>=0

- macro: open_directory
  condition: (evt.type=open or evt.type=openat) and evt.is_open_read=true and fd.typechar='d' and fd.num>=0

- macro: never_true
  condition: (evt.num=0)

- macro: always_true
  condition: (evt.num>=0)
```

システムコールイベントのドロップなどの場合、プロセス名に関する情報が見つからないことがあります。ファイルを開くなどのアクションを実行しているプロセスの身元に本当に依存するいくつかのルールでは、プロセス名を知る必要があります。

```
- macro: proc_name_exists
  condition: (proc.name!="<NA>")

- macro: rename
  condition: evt.type in (rename, renameat, renameat2)

- macro: mkdir
  condition: evt.type in (mkdir, mkdirat)

- macro: remove
  condition: evt.type in (rmdir, unlink, unlinkat)

- macro: modify
  condition: rename or remove

- macro: spawned_process
  condition: evt.type = execve and evt.dir=<

- macro: create_symlink
  condition: evt.type in (symlink, symlinkat) and evt.dir=<

- macro: chmod
  condition: (evt.type in (chmod, fchmod, fchmodat) and evt.dir=<)

# ファイルカテゴリー
- macro: bin_dir
  condition: fd.directory in (/bin, /sbin, /usr/bin, /usr/sbin)

- macro: bin_dir_mkdir
  condition: >
     (evt.arg.path startswith /bin/ or
```

```
    evt.arg.path startswith /sbin/ or
    evt.arg.path startswith /usr/bin/ or
    evt.arg.path startswith /usr/sbin/)
- macro: bin_dir_rename
  condition: >
    (evt.arg.path startswith /bin/ or
    evt.arg.path startswith /sbin/ or
    evt.arg.path startswith /usr/bin/ or
    evt.arg.path startswith /usr/sbin/ or
    evt.arg.name startswith /bin/ or
    evt.arg.name startswith /sbin/ or
    evt.arg.name startswith /usr/bin/ or
    evt.arg.name startswith /usr/sbin/ or
    evt.arg.oldpath startswith /bin/ or
    evt.arg.oldpath startswith /sbin/ or
    evt.arg.oldpath startswith /usr/bin/ or
    evt.arg.oldpath startswith /usr/sbin/ or
    evt.arg.newpath startswith /bin/ or
    evt.arg.newpath startswith /sbin/ or
    evt.arg.newpath startswith /usr/bin/ or
    evt.arg.newpath startswith /usr/sbin/)
- macro: etc_dir
  condition: fd.name startswith /etc/

# 直下の書き込み、または /root 以下の任意の場所への書き込みを検出します。
- macro: root_dir
  condition: (fd.directory=/ or fd.name startswith /root/)

- list: shell_binaries
  items: [ash, bash, csh, ksh, sh, tcsh, zsh, dash]

- list: ssh_binaries
  items: [
    sshd, sftp-server, ssh-agent,
    ssh, scp, sftp,
    ssh-keygen, ssh-keysign, ssh-keyscan, ssh-add
    ]

- list: shell_mgmt_binaries
  items: [add-shell, remove-shell]

- macro: shell_procs
  condition: proc.name in (shell_binaries)

- list: coreutils_binaries
  items: [
    truncate, sha1sum, numfmt, fmt, fold, uniq, cut, who,
    groups, csplit, sort, expand, printf, printenv, unlink, tee, chcon, stat,
    basename, split, nice, "yes", whoami, sha224sum, hostid, users, stdbuf,
    base64, unexpand, cksum, od, paste, nproc, pathchk, sha256sum, wc, test,
    comm, arch, du, factor, sha512sum, md5sum, tr, runcon, env, dirname,
    tsort, join, shuf, install, logname, pinky, nohup, expr, pr, tty, timeout,
    tail, "[", seq, sha384sum, nl, head, id, mkfifo, sum, dircolors, ptx, shred,
    tac, link, chroot, vdir, chown, touch, ls, dd, uname, "true", pwd, date,
    chgrp, chmod, mktemp, cat, mknod, sync, ln, "false", rm, mv, cp, echo,
```

```
    readlink, sleep, stty, mkdir, df, dir, rmdir, touch
    ]

# dpkg -L login | grep bin | xargs ls -ld | grep -v '^d' | awk '{print $9}' | xargs -L 1 basename | tr "\\n" ","
- list: login_binaries
  items: [
    login, systemd, '"(systemd)"', systemd-logind, su,
    nologin, faillog, lastlog, newgrp, sg
    ]

# dpkg -L passwd | grep bin | xargs ls -ld | grep -v '^d' | awk '{print $9}' | xargs -L 1 basename | tr "\\n"
","
- list: passwd_binaries
  items: [
    shadowconfig, grpck, pwunconv, grpconv, pwck,
    groupmod, vipw, pwconv, useradd, newusers, cppw, chpasswd, usermod,
    groupadd, groupdel, grpunconv, chgpasswd, userdel, chage, chsh,
    gpasswd, chfn, expiry, passwd, vigr, cpgr, adduser, addgroup, deluser, delgroup
    ]

# repoquery -l shadow-utils | grep bin | xargs ls -ld | grep -v '^d' |
#     awk '{print $9}' | xargs -L 1 basename | tr "\\n" ","
- list: shadowutils_binaries
  items: [
    chage, gpasswd, lastlog, newgrp, sg, adduser, deluser, chpasswd,
    groupadd, groupdel, addgroup, delgroup, groupmems, groupmod, grpck, grpconv, grpunconv,
    newusers, pwck, pwconv, pwunconv, useradd, userdel, usermod, vigr, vipw, unix_chkpwd
    ]

- list: sysdigcloud_binaries
  items: [setup-backend, dragent, sdchecks]

- list: docker_binaries
  items: [docker, dockerd, exe, docker-compose, docker-entrypoi, docker-runc-cur, docker-current,
dockerd-current]

- list: k8s_binaries
  items: [hyperkube, skydns, kube2sky, exechealthz, weave-net, loopback, bridge, openshift-sdn, openshift]

- list: lxd_binaries
  items: [lxd, lxcfs]

- list: http_server_binaries
  items: [nginx, httpd, httpd-foregroun, lighttpd, apache, apache2]

- list: db_server_binaries
  items: [mysqld, postgres, sqlplus]

- list: mysql_mgmt_binaries
  items: [mysql_install_d, mysql_ssl_rsa_s]

- list: postgres_mgmt_binaries
  items: [pg_dumpall, pg_ctl, pg_lsclusters, pg_ctlcluster]

- list: db_mgmt_binaries
```

```
  items: [mysql_mgmt_binaries, postgres_mgmt_binaries]

- list: nosql_server_binaries
  items: [couchdb, memcached, redis-server, rabbitmq-server, mongod]

- list: gitlab_binaries
  items: [gitlab-shell, gitlab-mon, gitlab-runner-b, git]

- list: interpreted_binaries
  items: [lua, node, perl, perl5, perl6, php, python, python2, python3, ruby, tcl]

- macro: interpreted_procs
  condition: >
    (proc.name in (interpreted_binaries))
- macro: server_procs
  condition: proc.name in (http_server_binaries, db_server_binaries, docker_binaries, sshd)
```

明示的な引用符は、フィルタ式で - 文字が解釈されないようにするために必要です。

```
- list: rpm_binaries
  items: [dnf, rpm, rpmkey, yum, '"75-system-updat"', rhsmcertd-worke, subscription-ma,
          repoquery, rpmkeys, rpmq, yum-cron, yum-config-mana, yum-debug-dump,
          abrt-action-sav, rpmdb_stat, microdnf, rhn_check, yumdb]

- list: openscap_rpm_binaries
  items: [probe_rpminfo, probe_rpmverify, probe_rpmverifyfile, probe_rpmverifypackage]

- macro: rpm_procs
  condition: (proc.name in (rpm_binaries, openscap_rpm_binaries) or proc.name in (salt-minion))

- list: deb_binaries
  items: [dpkg, dpkg-preconfigu, dpkg-reconfigur, dpkg-divert, apt, apt-get, aptitude,
    frontend, preinst, add-apt-reposit, apt-auto-remova, apt-key,
    apt-listchanges, unattended-upgr, apt-add-reposit, apt-config, apt-cache
    ]
```

切り捨てられた dpkg-preconfigu は意図的なもので、プロセス名は sysdig レベルで切り捨てられています。

```
- list: package_mgmt_binaries
  items: [rpm_binaries, deb_binaries, update-alternat, gem, pip, pip3, sane-utils.post, alternatives,
chef-client, apk, snapd]

- macro: package_mgmt_procs
  condition: proc.name in (package_mgmt_binaries)

- macro: package_mgmt_ancestor_procs
  condition: proc.pname in (package_mgmt_binaries) or
             proc.aname[2] in (package_mgmt_binaries) or
             proc.aname[3] in (package_mgmt_binaries) or
             proc.aname[4] in (package_mgmt_binaries)
```

```
- macro: coreos_write_ssh_dir
  condition: (proc.name=update-ssh-keys and fd.name startswith /home/core/.ssh)

- macro: run_by_package_mgmt_binaries
  condition: proc.aname in (package_mgmt_binaries, needrestart)

- list: ssl_mgmt_binaries
  items: [ca-certificates]

- list: dhcp_binaries
  items: [dhclient, dhclient-script, 11-dhclient]
```

異なる特権を持つ、または異なるユーザーとして他のプログラムを実行するプロセスの正規セット。

```
- list: userexec_binaries
  items: [sudo, su, suexec, critical-stack, dzdo]

- list: known_setuid_binaries
  items: [
    sshd, dbus-daemon-lau, ping, ping6, critical-stack-, pmmcli,
    filemng, PassengerAgent, bwrap, osdetect, nginxmng, sw-engine-fpm,
    start-stop-daem
    ]

- list: user_mgmt_binaries
  items: [login_binaries, passwd_binaries, shadowutils_binaries]

- list: dev_creation_binaries
  items: [blkid, rename_device, update_engine, sgdisk]

- list: hids_binaries
  items: [aide, aide.wrapper, update-aide.con, logcheck, syslog-summary, osqueryd, ossec-syscheckd]

- list: vpn_binaries
  items: [openvpn]

- list: nomachine_binaries
  items: [nxexec, nxnode.bin, nxserver.bin, nxclient.bin]

- macro: system_procs
  condition: proc.name in (coreutils_binaries, user_mgmt_binaries)

- list: mail_binaries
  items: [
    sendmail, sendmail-msp, postfix, procmail, exim4,
    pickup, showq, mailq, dovecot, imap-login, imap,
    mailmng-core, pop3-login, dovecot-lda, pop3
    ]

- list: mail_config_binaries
  items: [
```

```
    update_conf, parse_mc, makemap_hash, newaliases, update_mk, update_tlsm4,
    update_db, update_mc, ssmtp.postinst, mailq, postalias, postfix.config.,
    postfix.config, postfix-script, postconf
    ]

- list: sensitive_file_names
  items: [/etc/shadow, /etc/sudoers, /etc/pam.conf, /etc/security/pwquality.conf]

- list: sensitive_directory_names
  items: [/, /etc, /etc/, /root, /root/]

- macro: sensitive_files
  condition: >
    fd.name startswith /etc and
    (fd.name in (sensitive_file_names)
     or fd.directory in (/etc/sudoers.d, /etc/pam.d))
```

プロセスが新しいことを示します。プロセスが開始されてからの時間を使用して、5秒のしきい値を使用して現在検出されています。

```
- macro: proc_is_new
  condition: proc.duration <= 5000000000

# ネットワーク
- macro: inbound
  condition: >
    (((evt.type in (accept,listen) and evt.dir=<) or
      (evt.type in (recvfrom,recvmsg) and evt.dir=< and
       fd.l4proto != tcp and fd.connected=false and fd.name_changed=true)) and
     (fd.typechar = 4 or fd.typechar = 6) and
     (fd.ip != "0.0.0.0" and fd.net != "127.0.0.0/8") and
     (evt.rawres >= 0 or evt.res = EINPROGRESS))

# RFC1918のアドレスは、プライベートネットワークでの使用のために割り当てられています。
- list: rfc_1918_addresses
  items: ['"10.0.0.0/8"', '"172.16.0.0/12"', '"192.168.0.0/16"']

- macro: outbound
  condition: >
    (((evt.type = connect and evt.dir=<) or
      (evt.type in (sendto,sendmsg) and evt.dir=< and
       fd.l4proto != tcp and fd.connected=false and fd.name_changed=true)) and
     (fd.typechar = 4 or fd.typechar = 6) and
     (fd.ip != "0.0.0.0" and fd.net != "127.0.0.0/8" and not fd.snet in (rfc_1918_addresses)) and
     (evt.rawres >= 0 or evt.res = EINPROGRESS))

# インバウンド/アウトバウンドに非常に似ていますが、効率化のためにテストを組み合わせています。
- macro: inbound_outbound
  condition: >
    ((((evt.type in (accept,listen,connect) and evt.dir=<)) or
     (fd.typechar = 4 or fd.typechar = 6)) and
     (fd.ip != "0.0.0.0" and fd.net != "127.0.0.0/8") and
     (evt.rawres >= 0 or evt.res = EINPROGRESS))
```

```
- macro: ssh_port
  condition: fd.sport=22
```

Falcoルール - 1

1. **許可されていないSSH接続：Disallowed SSH Connection**
   許可されているホストグループ以外のホストへの新しい ssh 接続を検出する。
   ローカル/ユーザルールファイルでは、このマクロをオーバーライドして ssh 接続が許可されているサーバを列挙することができます。たとえば、ssh 接続が許可されている ssh ゲートウェイホストがあるかもしれません。メインの falco ルールファイルでは、ssh アクセスが許可されている特定のホストを知る方法はありませんので、このマクロは ssh_port を繰り返すだけで、事実上すべてのホストからの ssh を許可します。オーバーライドされたマクロでは、条件は次のようになります。 "fd.sip="a.b.c.d" or fd.sip="e.f.g.h" or ...."

```
- macro: allowed_ssh_hosts
  condition: ssh_port

- rule: Disallowed SSH Connection
  desc: Detect any new ssh connection to a host other than those in an allowed group of hosts
  condition: (inbound_outbound) and ssh_port and not allowed_ssh_hosts
  output: Disallowed SSH Connection (command=%proc.cmdline connection=%fd.name user=%user.name
user_loginuid=%user.loginuid container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_remote_service]
```

2. **予期しないアウトバウンド接続先：Unexpected outbound connection destination**
   許可されたIP、ネットワーク、またはドメイン名のセット外の宛先へのアウトバウンド接続を検出します。
   これらのルールとそれをサポートするマクロは、fd.*ip および fd.*ip.name フィールドを使用して、接続情報を ip、ネットマスク、および完全なドメイン名と一致させる方法の例です。このルールを使用するには、consider_all_outbound_connsを修正し、
   allowed_{source,destination}_{ipaddrs,network,domains}に環境に合った値を入力する必要があります。

```
- macro: consider_all_outbound_conns
  condition: (never_true)

# これは個々の IP またはネットマスクのいずれかであることに注意してください。
- list: allowed_outbound_destination_ipaddrs
  items: ['"127.0.0.1"', '"8.8.8.8"']

- list: allowed_outbound_destination_networks
  items: ['"127.0.0.1/8"']

- list: allowed_outbound_destination_domains
  items: [google.com, www.yahoo.com]

- rule: Unexpected outbound connection destination
  desc: Detect any outbound connection to a destination outside of an allowed set of ips, networks, or domain
names
  condition: >
    consider_all_outbound_conns and outbound and not
    ((fd.sip in (allowed_outbound_destination_ipaddrs)) or
     (fd.snet in (allowed_outbound_destination_networks)) or
     (fd.sip.name in (allowed_outbound_destination_domains)))
  output: Disallowed outbound connection destination (command=%proc.cmdline connection=%fd.name user=%user.name
user_loginuid=%user.loginuid container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network]

- macro: consider_all_inbound_conns
  condition: (never_true)

- list: allowed_inbound_source_ipaddrs
  items: ['"127.0.0.1"']

- list: allowed_inbound_source_networks
  items: ['"127.0.0.1/8"', '"10.0.0.0/8"']

- list: allowed_inbound_source_domains
  items: [google.com]
```

3. 予期しないインバウンド接続のソース：Unexpected inbound connection source

許可されているip、ネットワーク、ドメイン名のセット外のソースからの受信接続を検出します。

```
- rule: Unexpected inbound connection source
  desc: Detect any inbound connection from a source outside of an allowed set of ips, networks, or domain names
  condition: >
    consider_all_inbound_conns and inbound and not
    ((fd.cip in (allowed_inbound_source_ipaddrs)) or
     (fd.cnet in (allowed_inbound_source_networks)) or
     (fd.cip.name in (allowed_inbound_source_domains)))
  output: Disallowed inbound connection source (command=%proc.cmdline connection=%fd.name user=%user.name
user_loginuid=%user.loginuid container_id=%container.id image=%container.image.repository)
  priority: NOTICE
```

```
  tags: [network]

- list: bash_config_filenames
  items: [.bashrc, .bash_profile, .bash_history, .bash_login, .bash_logout, .inputrc, .profile]

- list: bash_config_files
  items: [/etc/profile, /etc/bashrc]

# Covers both csh and tcsh
- list: csh_config_filenames
  items: [.cshrc, .login, .logout, .history, .tcshrc, .cshdirs]

- list: csh_config_files
  items: [/etc/csh.cshrc, /etc/csh.login]

- list: zsh_config_filenames
  items: [.zshenv, .zprofile, .zshrc, .zlogin, .zlogout]

- list: shell_config_filenames
  items: [bash_config_filenames, csh_config_filenames, zsh_config_filenames]

- list: shell_config_files
  items: [bash_config_files, csh_config_files]

- list: shell_config_directories
  items: [/etc/zsh]
```

4. シェル設定ファイルの変更：Modify Shell Configuration File

シェル設定ファイルを変更しようとする試みを検出

```
- rule: Modify Shell Configuration File
  desc: Detect attempt to modify shell configuration files
  condition: >
    open_write and
    (fd.filename in (shell_config_filenames) or
     fd.name in (shell_config_files) or
     fd.directory in (shell_config_directories))
    and not proc.name in (shell_binaries)
    and not exe_running_docker_save
  output: >
    a shell configuration file has been modified (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline pcmdline=%proc.pcmdline file=%fd.name container_id=%container.id
image=%container.image.repository)
  priority:
    WARNING
  tags: [file, mitre_persistence]
```

5. **シェル設定ファイルの読み込み：Read Shell Configuration File**

   シェル以外のプログラムによるシェル設定ファイルの読み取りの試みの検出

   シェル設定ファイルを正規にリードするものがたくさんあるので、このルールはデフォルトでは有効になっていません。有効にしたい場合は、- macro: consider_shell_config_reads を修正してください。

```
# This rule is not enabled by default, as there are many legitimate
# readers of shell config files. If you want to enable it, modify the
# following macro.

- macro: consider_shell_config_reads
  condition: (never_true)

- rule: Read Shell Configuration File
  desc: Detect attempts to read shell configuration files by non-shell programs
  condition: >
    open_read and
    consider_shell_config_reads and
    (fd.filename in (shell_config_filenames) or
     fd.name in (shell_config_files) or
     fd.directory in (shell_config_directories)) and
    (not proc.name in (shell_binaries))
  output: >
    a shell configuration file was read by a non-shell program (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline file=%fd.name container_id=%container.id image=%container.image.repository)
  priority:
    WARNING
  tags: [file, mitre_discovery]

- macro: consider_all_cron_jobs
  condition: (never_true)

- macro: user_known_cron_jobs
  condition: (never_true)

- rule: Schedule Cron Jobs
  desc: Detect cron jobs scheduled
  condition: >
    ((open_write and fd.name startswith /etc/cron) or
     (spawned_process and proc.name = "crontab")) and
    consider_all_cron_jobs and
    not user_known_cron_jobs
  output: >
    Cron jobs were scheduled to run (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
    file=%fd.name container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority:
    NOTICE
  tags: [file, mitre_persistence]
```

6. **Cronジョブのスケジュール：Schedule Cron Jobs**
   スケジュールされたcronジョブの検出

```
- macro: consider_all_cron_jobs
  condition: (never_true)

- macro: user_known_cron_jobs
  condition: (never_true)

- rule: Schedule Cron Jobs
  desc: Detect cron jobs scheduled
  condition: >
    ((open_write and fd.name startswith /etc/cron) or
     (spawned_process and proc.name = "crontab")) and
    consider_all_cron_jobs and
    not user_known_cron_jobs
  output: >
    Cron jobs were scheduled to run (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
    file=%fd.name container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority:
    NOTICE
  tags: [file, mitre_persistence]
```

共通マクロ、リスト - 2

イベントがコンテナ内で発生したかどうかをテストします。
出力フィールドにコンテナ情報を表示する場合は、%container.infoを使用し、先頭の用語を使用しないでください (file=%fd.name %container.info user=%user.name user_loginuid=%user.loginuidではなく、file=%fd.name container=%container.info user=%user.name user_loginuid=%user.loginuid)。コマンドラインで-pk/pm/pcが指定されているかどうかと、コンテキストに応じて出力が変わります。

```
- macro: container
  condition: (container.id != host)

- macro: container_started
  condition: >
    ((evt.type = container or
     (spawned_process and proc.vpid=1)) and
     container.image.repository != incomplete)
- macro: interactive
  condition: >
    ((proc.aname=sshd and proc.name != sshd) or
    proc.name=systemd-logind or proc.name=login)
- list: cron_binaries
  items: [anacron, cron, crond, crontab]
```

```
# https://github.com/liske/needrestart
- list: needrestart_binaries
  items: [needrestart, 10-dpkg, 20-rpm, 30-pacman]

# sshkit で実行可能なスクリプト
- list: sshkit_script_binaries
  items: [10_etc_sudoers., 10_passwd_group]

- list: plesk_binaries
  items: [sw-engine, sw-engine-fpm, sw-engine-kv, filemng, f2bmng]
```

絶対にログインしてはいけないシステムユーザ。独自のサービスユーザ (例えば 'apache' や 'mysqld' など) をここに追加することを検討してください。

```
- macro: system_users
  condition: user.name in (bin, daemon, games, lp, mail, nobody, sshd, sync, uucp, www-data)
```

これらのマクロはすぐに削除されます。広く使われているルールファイルとの互換性を維持するためだけに残しています。

```
# 開始点 非推奨
- macro: parent_ansible_running_python
  condition: (proc.pname in (python, pypy, python3) and proc.pcmdline contains ansible)

- macro: parent_bro_running_python
  condition: (proc.pname=python and proc.cmdline contains /usr/share/broctl)

- macro: parent_python_running_denyhosts
  condition: >
    (proc.cmdline startswith "denyhosts.py /usr/bin/denyhosts.py" or
     (proc.pname=python and
     (proc.pcmdline contains /usr/sbin/denyhosts or
      proc.pcmdline contains /usr/local/bin/denyhosts.py)))
- macro: parent_python_running_sdchecks
  condition: >
    (proc.pname in (python, python2.7) and
    (proc.pcmdline contains /opt/draios/bin/sdchecks))
- macro: python_running_sdchecks
  condition: >
    (proc.name in (python, python2.7) and
    (proc.cmdline contains /opt/draios/bin/sdchecks))
- macro: parent_linux_image_upgrade_script
  condition: proc.pname startswith linux-image-

- macro: parent_java_running_echo
  condition: (proc.pname=java and proc.cmdline startswith "sh -c echo")

- macro: parent_scripting_running_builds
  condition: >
```

```
    (proc.pname in (php,php5-fpm,php-fpm7.1,python,ruby,ruby2.3,ruby2.1,node,conda) and (
       proc.cmdline startswith "sh -c git" or
       proc.cmdline startswith "sh -c date" or
       proc.cmdline startswith "sh -c /usr/bin/g++" or
       proc.cmdline startswith "sh -c /usr/bin/gcc" or
       proc.cmdline startswith "sh -c gcc" or
       proc.cmdline startswith "sh -c if type gcc" or
       proc.cmdline startswith "sh -c cd '/var/www/edi/';LC_ALL=en_US.UTF-8 git" or
       proc.cmdline startswith "sh -c /var/www/edi/bin/sftp.sh" or
       proc.cmdline startswith "sh -c /usr/src/app/crxlsx/bin/linux/crxlsx" or
       proc.cmdline startswith "sh -c make parent" or
       proc.cmdline startswith "node /jenkins/tools" or
       proc.cmdline startswith "sh -c '/usr/bin/node'" or
       proc.cmdline startswith "sh -c stty -a |" or
       proc.pcmdline startswith "node /opt/nodejs/bin/yarn" or
       proc.pcmdline startswith "node /usr/local/bin/yarn" or
       proc.pcmdline startswith "node /root/.config/yarn" or
       proc.pcmdline startswith "node /opt/yarn/bin/yarn.js"))
- macro: httpd_writing_ssl_conf
  condition: >
    (proc.pname=run-httpd and
     (proc.cmdline startswith "sed -ri" or proc.cmdline startswith "sed -i") and
     (fd.name startswith /etc/httpd/conf.d/ or fd.name startswith /etc/httpd/conf))
- macro: userhelper_writing_etc_security
  condition: (proc.name=userhelper and fd.name startswith /etc/security)

- macro: parent_Xvfb_running_xkbcomp
  condition: (proc.pname=Xvfb and proc.cmdline startswith 'sh -c "/usr/bin/xkbcomp"')

- macro: parent_nginx_running_serf
  condition: (proc.pname=nginx and proc.cmdline startswith "sh -c serf")

- macro: parent_node_running_npm
  condition: (proc.pcmdline startswith "node /usr/local/bin/npm" or
              proc.pcmdline startswith "node /usr/local/nodejs/bin/npm" or
              proc.pcmdline startswith "node /opt/rh/rh-nodejs6/root/usr/bin/npm")

- macro: parent_java_running_sbt
  condition: (proc.pname=java and proc.pcmdline contains sbt-launch.jar)

- list: known_container_shell_spawn_cmdlines
  items: []

- list: known_shell_spawn_binaries
  items: []

## 終了点 非推奨



- macro: ansible_running_python
  condition: (proc.name in (python, pypy, python3) and proc.cmdline contains ansible)

- macro: python_running_chef
```

```
  condition: (proc.name=python and (proc.cmdline contains yum-dump.py or proc.cmdline="python
/usr/bin/chef-monitor.py"))

- macro: python_running_denyhosts
  condition: >
    (proc.name=python and
    (proc.cmdline contains /usr/sbin/denyhosts or
     proc.cmdline contains /usr/local/bin/denyhosts.py))
```

Qualysは様々なレベルのシェルサブプロセスを実行しているようです。これは、完全な親階層を横断する proc.aname のコストをかけずに、いくつかのレベルでチェックします。

```
- macro: run_by_qualys
  condition: >
    (proc.pname=qualys-cloud-ag or
     proc.aname[2]=qualys-cloud-ag or
     proc.aname[3]=qualys-cloud-ag or
     proc.aname[4]=qualys-cloud-ag)
- macro: run_by_sumologic_securefiles
  condition: >
    ((proc.cmdline="usermod -a -G sumologic_collector" or
      proc.cmdline="groupadd sumologic_collector") and
     (proc.pname=secureFiles.sh and proc.aname[2]=java))
- macro: run_by_yum
  condition: ((proc.pname=sh and proc.aname[2]=yum) or
              (proc.aname[2]=sh and proc.aname[3]=yum))

- macro: run_by_ms_oms
  condition: >
    (proc.aname[3] startswith omsagent- or
     proc.aname[3] startswith scx-)
- macro: run_by_google_accounts_daemon
  condition: >
    (proc.aname[1] startswith google_accounts or
     proc.aname[2] startswith google_accounts or
     proc.aname[3] startswith google_accounts)
```

chefも同様に様々なレベルのシェルサブプロセスを実行しているようです。

```
- macro: run_by_chef
  condition: (proc.aname[2]=chef_command_wr or proc.aname[3]=chef_command_wr or
              proc.aname[2]=chef-client or proc.aname[3]=chef-client or
              proc.name=chef-client)

- macro: run_by_adclient
  condition: (proc.aname[2]=adclient or proc.aname[3]=adclient or proc.aname[4]=adclient)

- macro: run_by_centrify
  condition: (proc.aname[2]=centrify or proc.aname[3]=centrify or proc.aname[4]=centrify)

- macro: run_by_puppet
```

```
  condition: (proc.aname[2]=puppet or proc.aname[3]=puppet)
```

また、scl を介したセミインダイレクトな実行にも対応しています。

```
- macro: run_by_foreman
  condition: >
    (user.name=foreman and
     ((proc.pname in (rake, ruby, scl) and proc.aname[5] in (tfm-rake,tfm-ruby)) or
     (proc.pname=scl and proc.aname[2] in (tfm-rake,tfm-ruby))))
- macro: java_running_sdjagent
  condition: proc.name=java and proc.cmdline contains sdjagent.jar

- macro: kubelet_running_loopback
  condition: (proc.pname=kubelet and proc.name=loopback)

- macro: python_mesos_marathon_scripting
  condition: (proc.pcmdline startswith "python3 /marathon-lb/marathon_lb.py")

- macro: splunk_running_forwarder
  condition: (proc.pname=splunkd and proc.cmdline startswith "sh -c /opt/splunkforwarder")

- macro: parent_supervise_running_multilog
  condition: (proc.name=multilog and proc.pname=supervise)

- macro: supervise_writing_status
  condition: (proc.name in (supervise,svc) and fd.name startswith "/etc/sb/")

- macro: pki_realm_writing_realms
  condition: (proc.cmdline startswith "bash /usr/local/lib/pki/pki-realm" and fd.name startswith
/etc/pki/realms)

- macro: htpasswd_writing_passwd
  condition: (proc.name=htpasswd and fd.name=/etc/nginx/.htpasswd)

- macro: lvprogs_writing_conf
  condition: >
    (proc.name in (dmeventd,lvcreate,pvscan,lvs) and
     (fd.name startswith /etc/lvm/archive or
      fd.name startswith /etc/lvm/backup or
      fd.name startswith /etc/lvm/cache))
- macro: ovsdb_writing_openvswitch
  condition: (proc.name=ovsdb-server and fd.directory=/etc/openvswitch)

- macro: perl_running_plesk
  condition: (proc.cmdline startswith "perl /opt/psa/admin/bin/plesk_agent_manager" or
              proc.pcmdline startswith "perl /opt/psa/admin/bin/plesk_agent_manager")

- macro: perl_running_updmap
  condition: (proc.cmdline startswith "perl /usr/bin/updmap")

- macro: perl_running_centrifydc
  condition: (proc.cmdline startswith "perl /usr/share/centrifydc")
```

```
- macro: runuser_reading_pam
  condition: (proc.name=runuser and fd.directory=/etc/pam.d)
```

CIS Linuxベンチマークプログラム

```
- macro: linux_bench_reading_etc_shadow
  condition: ((proc.aname[2]=linux-bench and
               proc.name in (awk,cut,grep)) and
              (fd.name=/etc/shadow or
               fd.directory=/etc/pam.d))

- macro: parent_ucf_writing_conf
  condition: (proc.pname=ucf and proc.aname[2]=frontend)

- macro: consul_template_writing_conf
  condition: >
    ((proc.name=consul-template and fd.name startswith /etc/haproxy) or
     (proc.name=reload.sh and proc.aname[2]=consul-template and fd.name startswith /etc/ssl))
- macro: countly_writing_nginx_conf
  condition: (proc.cmdline startswith "nodejs /opt/countly/bin" and fd.name startswith /etc/nginx)

- list: ms_oms_binaries
  items: [omi.postinst, omsconfig.posti, scx.postinst, omsadmin.sh, omiagent]

- macro: ms_oms_writing_conf
  condition: >
    ((proc.name in (omiagent,omsagent,in_heartbeat_r*,omsadmin.sh,PerformInventor,dsc_host)
       or proc.pname in (ms_oms_binaries)
       or proc.aname[2] in (ms_oms_binaries))
     and (fd.name startswith /etc/opt/omi or fd.name startswith /etc/opt/microsoft/omsagent))
- macro: ms_scx_writing_conf
  condition: (proc.name in (GetLinuxOS.sh) and fd.name startswith /etc/opt/microsoft/scx)

- macro: azure_scripts_writing_conf
  condition: (proc.pname startswith "bash /var/lib/waagent/" and fd.name startswith /etc/azure)

- macro: azure_networkwatcher_writing_conf
  condition: (proc.name in (NetworkWatcherA) and fd.name=/etc/init.d/AzureNetworkWatcherAgent)

- macro: couchdb_writing_conf
  condition: (proc.name=beam.smp and proc.cmdline contains couchdb and fd.name startswith /etc/couchdb)

- macro: update_texmf_writing_conf
  condition: (proc.name=update-texmf and fd.name startswith /etc/texmf)

- macro: slapadd_writing_conf
  condition: (proc.name=slapadd and fd.name startswith /etc/ldap)

- macro: openldap_writing_conf
  condition: (proc.pname=run-openldap.sh and fd.name startswith /etc/openldap)

- macro: ucpagent_writing_conf
```

```
  condition: (proc.name=apiserver and container.image.repository=docker/ucp-agent and
fd.name=/etc/authorization_config.cfg)

- macro: iscsi_writing_conf
  condition: (proc.name=iscsiadm and fd.name startswith /etc/iscsi)

- macro: istio_writing_conf
  condition: (proc.name=pilot-agent and fd.name startswith /etc/istio)

- macro: symantec_writing_conf
  condition: >
    ((proc.name=symcfgd and fd.name startswith /etc/symantec) or
     (proc.name=navdefutil and fd.name=/etc/symc-defutils.conf))
- macro: liveupdate_writing_conf
  condition: (proc.cmdline startswith "java LiveUpdate" and fd.name in (/etc/liveupdate.conf,
/etc/Product.Catalog.JavaLiveUpdate))

- macro: rancher_agent
  condition: (proc.name=agent and container.image.repository contains "rancher/agent")

- macro: rancher_network_manager
  condition: (proc.name=rancher-bridge and container.image.repository contains "rancher/network-manager")

- macro: sosreport_writing_files
  condition: >
    (proc.name=urlgrabber-ext- and proc.aname[3]=sosreport and
     (fd.name startswith /etc/pkt/nssdb or fd.name startswith /etc/pki/nssdb))
- macro: pkgmgmt_progs_writing_pki
  condition: >
    (proc.name=urlgrabber-ext- and proc.pname in (yum, yum-cron, repoquery) and
     (fd.name startswith /etc/pkt/nssdb or fd.name startswith /etc/pki/nssdb))
- macro: update_ca_trust_writing_pki
  condition: (proc.pname=update-ca-trust and proc.name=trust and fd.name startswith /etc/pki)

- macro: brandbot_writing_os_release
  condition: proc.name=brandbot and fd.name=/etc/os-release

- macro: selinux_writing_conf
  condition: (proc.name in (semodule,genhomedircon,sefcontext_comp) and fd.name startswith /etc/selinux)

- list: veritas_binaries
  items: [vxconfigd, sfcache, vxclustadm, vxdctl, vxprint, vxdmpadm, vxdisk, vxdg, vxassist, vxtune]

- macro: veritas_driver_script
  condition: (proc.cmdline startswith "perl /opt/VRTSsfmh/bin/mh_driver.pl")

- macro: veritas_progs
  condition: (proc.name in (veritas_binaries) or veritas_driver_script)

- macro: veritas_writing_config
  condition: (veritas_progs and (fd.name startswith /etc/vx or fd.name startswith /etc/opt/VRTS or fd.name
startswith /etc/vom))

- macro: nginx_writing_conf
  condition: (proc.name in (nginx,nginx-ingress-c,nginx-ingress) and (fd.name startswith /etc/nginx or fd.name
```

```
startswith /etc/ingress-controller))

- macro: nginx_writing_certs
  condition: >
    (((proc.name=openssl and proc.pname=nginx-launch.sh) or proc.name=nginx-launch.sh) and fd.name startswith
/etc/nginx/certs)
- macro: chef_client_writing_conf
  condition: (proc.pcmdline startswith "chef-client /opt/gitlab" and fd.name startswith /etc/gitlab)

- macro: centrify_writing_krb
  condition: (proc.name in (adjoin,addns) and fd.name startswith /etc/krb5)

- macro: cockpit_writing_conf
  condition: >
    ((proc.pname=cockpit-kube-la or proc.aname[2]=cockpit-kube-la)
     and fd.name startswith /etc/cockpit)
- macro: ipsec_writing_conf
  condition: (proc.name=start-ipsec.sh and fd.directory=/etc/ipsec)

- macro: exe_running_docker_save
  condition: >
    proc.name = "exe"
    and (proc.cmdline contains "/var/lib/docker"
    or proc.cmdline contains "/var/run/docker")
    and proc.pname in (dockerd, docker, dockerd-current, docker-current)
```

理想的にはここにも長さチェックがあるといいのですが、sysdig のフィルタチェックには len() のような演算子がありません。

```
- macro: sed_temporary_file
  condition: (proc.name=sed and fd.name startswith "/etc/sed")

- macro: python_running_get_pip
  condition: (proc.cmdline startswith "python get-pip.py")

- macro: python_running_ms_oms
  condition: (proc.cmdline startswith "python /var/lib/waagent/")

- macro: gugent_writing_guestagent_log
  condition: (proc.name=gugent and fd.name=GuestAgent.log)

- macro: dse_writing_tmp
  condition: (proc.name=dse-entrypoint and fd.name=/root/tmp__)

- macro: zap_writing_state
  condition: (proc.name=java and proc.cmdline contains "jar /zap" and fd.name startswith /root/.ZAP)

- macro: airflow_writing_state
  condition: (proc.name=airflow and fd.name startswith /root/airflow)

- macro: rpm_writing_root_rpmdb
  condition: (proc.name=rpm and fd.directory=/root/.rpmdb)

- macro: maven_writing_groovy
```

```
  condition: (proc.name=java and proc.cmdline contains "classpath /usr/local/apache-maven" and fd.name
startswith /root/.groovy)

- macro: chef_writing_conf
  condition: (proc.name=chef-client and fd.name startswith /root/.chef)

- macro: kubectl_writing_state
  condition: (proc.name in (kubectl,oc) and fd.name startswith /root/.kube)

- macro: java_running_cassandra
  condition: (proc.name=java and proc.cmdline contains "cassandra.jar")

- macro: cassandra_writing_state
  condition: (java_running_cassandra and fd.directory=/root/.cassandra)
```

Istio、calicoに関するマクロ、リスト

```
- macro: galley_writing_state
  condition: (proc.name=galley and fd.name in (known_istio_files))

- list: known_istio_files
  items: [/healthready, /healthliveness]

- macro: calico_writing_state
  condition: (proc.name=kube-controller and fd.name startswith /status.json and k8s.pod.name startswith calico)

- macro: calico_writing_envvars
  condition: (proc.name=start_runit and fd.name startswith "/etc/envvars" and container.image.repository
endswith "calico/node")
```

Falcoルール - 2

7. **パッケージリポジトリの更新：Update Package Repository**
   パッケージリポジトリの更新を検出します。

```
- list: repository_files
  items: [sources.list]

- list: repository_directories
  items: [/etc/apt/sources.list.d, /etc/yum.repos.d, /etc/apt]

- macro: access_repositories
  condition: (fd.directory in (repository_directories) or
              (fd.name pmatch (repository_directories) and
               fd.filename in (repository_files)))
```

```
- macro: modify_repositories
  condition: (evt.arg.newpath pmatch (repository_directories))

- macro: user_known_update_package_registry
  condition: (never_true)

- rule: Update Package Repository
  desc: Detect package repositories get updated
  condition: >
    ((open_write and access_repositories) or (modify and modify_repositories))
    and not package_mgmt_procs
    and not package_mgmt_ancestor_procs
    and not exe_running_docker_save
    and not user_known_update_package_registry
  output: >
    Repository files get updated (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
pcmdline=%proc.pcmdline file=%fd.name newpath=%evt.arg.newpath container_id=%container.id
image=%container.image.repository)
  priority:
    NOTICE
  tags: [filesystem, mitre_persistence]
```

8. **バイナリーディレクトリ以下への書き込み：Write below binary dir**

   バイナリディレクトリのセット以下のファイルへの書き込みを試みます。

   ユーザーはこのマクロを上書きして、バイナリディレクトリへの書き込みが無視される条件を指定する必要があります。例えば、ci/cd ビルドのコンテキストでバイナリをインストールしても問題ないかもしれません。

```
- macro: user_known_write_below_binary_dir_activities
  condition: (never_true)

- rule: Write below binary dir
  desc: an attempt to write to any file below a set of binary directories
  condition: >
    bin_dir and evt.dir = < and open_write
    and not package_mgmt_procs
    and not exe_running_docker_save
    and not python_running_get_pip
    and not python_running_ms_oms
    and not user_known_write_below_binary_dir_activities
  output: >
    File below a known binary directory opened for writing (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline file=%fd.name parent=%proc.pname pcmdline=%proc.pcmdline gparent=%proc.aname[2]
container_id=%container.id image=%container.image.repository)
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

バイナリディレクトリの下に書き込むというルールでカバーされているディレクトリの上に、より広いディレクトリのセットを一般的に監視したい場合は、以下のルールとリストを使用することができます。

```
- list: monitored_directories
  items: [/boot, /lib, /lib64, /usr/lib, /usr/local/lib, /usr/local/sbin, /usr/local/bin, /root/.ssh,
/etc/cardserver]
```

https://github.com/draios/sysdig/pull/1153 まで https://github.com/draios/sysdig/issues/1152 を修正したものが広く公開されています。
パス名のマッチに glob 演算子を使うことはできません。それまでは、ssh ディレクトリにマッチするかどうかのチェックを緩くしています。修正されたら、"fd.name glob '/home/*/.ssh/*'" を使うようになります。

```
- macro: user_ssh_directory
  condition: (fd.name startswith '/home' and fd.name contains '.ssh')

# google_accounts_(daemon)
- macro: google_accounts_daemon_writing_ssh
  condition: (proc.name=google_accounts and user_ssh_directory)

- macro: cloud_init_writing_ssh
  condition: (proc.name=cloud-init and user_ssh_directory)

- macro: mkinitramfs_writing_boot
  condition: (proc.pname in (mkinitramfs, update-initramf) and fd.directory=/boot)

- macro: monitored_dir
  condition: >
    (fd.directory in (monitored_directories)
     or user_ssh_directory)
    and not mkinitramfs_writing_boot
```

9. **監視対象ディレクトリへの書き込み：Write below monitored dir**
   バイナリディレクトリのセット以下のファイルへの書き込みを試みます。

   このマクロに条件を追加して (おそらく別ファイルで、このマクロを上書きして)、監視対象のディレクトリ以下に書き込むプログラムの特定の組み合わせを許可します。デフォルト値は常に false の式で、ルールの "not ...." が適用されると true になります。

```
- macro: user_known_write_monitored_dir_conditions
  condition: (never_true)

- rule: Write below monitored dir
```

```
  desc: an attempt to write to any file below a set of binary directories
  condition: >
    evt.dir = < and open_write and monitored_dir
    and not package_mgmt_procs
    and not coreos_write_ssh_dir
    and not exe_running_docker_save
    and not python_running_get_pip
    and not python_running_ms_oms
    and not google_accounts_daemon_writing_ssh
    and not cloud_init_writing_ssh
    and not user_known_write_monitored_dir_conditions
  output: >
    File below a monitored directory opened for writing (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline file=%fd.name parent=%proc.pname pcmdline=%proc.pcmdline gparent=%proc.aname[2]
container_id=%container.id image=%container.image.repository)
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

10. sshの情報をリードする：Read ssh information

ssh ディレクトリ以下のファイルを、ssh 以外のプログラムで読み込もうとした場合に検出
ansible などの多くのシステム管理ツールがこれらのファイル/パスを読み取ることができるため、
このルールはデフォルトでは無効になっています。このマクロを使用して有効にします。

```
- macro: consider_ssh_reads
  condition: (never_true)

- macro: user_known_read_ssh_information_activities
  condition: (never_true)

- rule: Read ssh information
  desc: Any attempt to read files below ssh directories by non-ssh programs
  condition: >
    ((open_read or open_directory) and
     consider_ssh_reads and
     (user_ssh_directory or fd.name startswith /root/.ssh) and
     not user_known_read_ssh_information_activities and
     not proc.name in (ssh_binaries))
  output: >
    ssh-related file/directory read by non-ssh program (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline file=%fd.name parent=%proc.pname pcmdline=%proc.pcmdline container_id=%container.id
image=%container.image.repository)
  priority: ERROR
  tags: [filesystem, mitre_discovery]
```

共通マクロ、リスト - 3

```
- list: safe_etc_dirs
  items: [/etc/cassandra, /etc/ssl/certs/java, /etc/logstash, /etc/nginx/conf.d, /etc/container_environment,
/etc/hrmconfig, /etc/fluent/configs.d]
```

```
- macro: fluentd_writing_conf_files
  condition: (proc.name=start-fluentd and fd.name in (/etc/fluent/fluent.conf, /etc/td-agent/td-agent.conf))

- macro: qualys_writing_conf_files
  condition: (proc.name=qualys-cloud-ag and fd.name=/etc/qualys/cloud-agent/qagent-log.conf)

- macro: git_writing_nssdb
  condition: (proc.name=git-remote-http and fd.directory=/etc/pki/nssdb)

- macro: plesk_writing_keys
  condition: (proc.name in (plesk_binaries) and fd.name startswith /etc/sw/keys)

- macro: plesk_install_writing_apache_conf
  condition: (proc.cmdline startswith "bash -hB /usr/lib/plesk-9.0/services/webserver.apache configure"
              and fd.name="/etc/apache2/apache2.conf.tmp")

- macro: plesk_running_mktemp
  condition: (proc.name=mktemp and proc.aname[3] in (plesk_binaries))

- macro: networkmanager_writing_resolv_conf
  condition: proc.aname[2]=nm-dispatcher and fd.name=/etc/resolv.conf

- macro: add_shell_writing_shells_tmp
  condition: (proc.name=add-shell and fd.name=/etc/shells.tmp)

- macro: duply_writing_exclude_files
  condition: (proc.name=touch and proc.pcmdline startswith "bash /usr/bin/duply" and fd.name startswith
"/etc/duply")

- macro: xmlcatalog_writing_files
  condition: (proc.name=update-xmlcatal and fd.directory=/etc/xml)

- macro: datadog_writing_conf
  condition: ((proc.cmdline startswith "python /opt/datadog-agent" or
               proc.cmdline startswith "entrypoint.sh /entrypoint.sh datadog start" or
               proc.cmdline startswith "agent.py /opt/datadog-agent")
              and fd.name startswith "/etc/dd-agent")

- macro: rancher_writing_conf
  condition: ((proc.name in (healthcheck, lb-controller, rancher-dns)) and
              (container.image.repository contains "rancher/healthcheck" or
               container.image.repository contains "rancher/lb-service-haproxy" or
               container.image.repository contains "rancher/dns") and
              (fd.name startswith "/etc/haproxy" or fd.name startswith "/etc/rancher-dns"))

- macro: rancher_writing_root
  condition: (proc.name=rancher-metadat and
              (container.image.repository contains "rancher/metadata" or container.image.repository contains
"rancher/lb-service-haproxy") and
              fd.name startswith "/answers.json")

- macro: checkpoint_writing_state
  condition: (proc.name=checkpoint and
              container.image.repository contains "coreos/pod-checkpointer" and
```

```
            fd.name startswith "/etc/kubernetes")

- macro: jboss_in_container_writing_passwd
  condition: >
    ((proc.cmdline="run-java.sh /opt/jboss/container/java/run/run-java.sh"
      or proc.cmdline="run-java.sh /opt/run-java/run-java.sh")
     and container
     and fd.name=/etc/passwd)
- macro: curl_writing_pki_db
  condition: (proc.name=curl and fd.directory=/etc/pki/nssdb)

- macro: haproxy_writing_conf
  condition: ((proc.name in (update-haproxy-,haproxy_reload.) or proc.pname in
(update-haproxy-,haproxy_reload,haproxy_reload.))
              and (fd.name=/etc/openvpn/client.map or fd.name startswith /etc/haproxy))

- macro: java_writing_conf
  condition: (proc.name=java and fd.name=/etc/.java/.systemPrefs/.system.lock)

- macro: rabbitmq_writing_conf
  condition: (proc.name=rabbitmq-server and fd.directory=/etc/rabbitmq)

- macro: rook_writing_conf
  condition: (proc.name=toolbox.sh and container.image.repository=rook/toolbox
             and fd.directory=/etc/ceph)

- macro: httpd_writing_conf_logs
  condition: (proc.name=httpd and fd.name startswith /etc/httpd/)

- macro: mysql_writing_conf
  condition: >
    ((proc.name in (start-mysql.sh, run-mysqld) or proc.pname=start-mysql.sh) and
     (fd.name startswith /etc/mysql or fd.directory=/etc/my.cnf.d))
- macro: redis_writing_conf
  condition: >
    (proc.name in (run-redis, redis-launcher.) and (fd.name=/etc/redis.conf or fd.name startswith /etc/redis))
- macro: openvpn_writing_conf
  condition: (proc.name in (openvpn,openvpn-entrypo) and fd.name startswith /etc/openvpn)

- macro: php_handlers_writing_conf
  condition: (proc.name=php_handlers_co and fd.name=/etc/psa/php_versions.json)

- macro: sed_writing_temp_file
  condition: >
    ((proc.aname[3]=cron_start.sh and fd.name startswith /etc/security/sed) or
     (proc.name=sed and (fd.name startswith /etc/apt/sources.list.d/sed or
                         fd.name startswith /etc/apt/sed or
                         fd.name startswith /etc/apt/apt.conf.d/sed)))
- macro: cron_start_writing_pam_env
  condition: (proc.cmdline="bash /usr/sbin/start-cron" and fd.name=/etc/security/pam_env.conf)
```

場合によっては、/etc を修正するコマンドを実行することもあります。

パッケージ管理プログラムのフルセットはまだ置いていません。

```
- macro: dpkg_scripting
  condition: (proc.aname[2] in (dpkg-reconfigur, dpkg-preconfigu))

- macro: ufw_writing_conf
  condition: (proc.name=ufw and fd.directory=/etc/ufw)

- macro: calico_writing_conf
  condition: >
    (((proc.name = calico-node) or
      (container.image.repository=gcr.io/projectcalico-org/node and proc.name in (start_runit, cp)) or
      (container.image.repository=gcr.io/projectcalico-org/cni and proc.name=sed))
     and fd.name startswith /etc/calico)
- macro: prometheus_conf_writing_conf
  condition: (proc.name=prometheus-conf and fd.name startswith /etc/prometheus/config_out)

- macro: openshift_writing_conf
  condition: (proc.name=oc and fd.name startswith /etc/origin/node)

- macro: keepalived_writing_conf
  condition: (proc.name=keepalived and fd.name=/etc/keepalived/keepalived.conf)

- macro: etcd_manager_updating_dns
  condition: (container and proc.name=etcd-manager and fd.name=/etc/hosts)

- macro: automount_using_mtab
  condition: (proc.pname = automount and fd.name startswith /etc/mtab)

- macro: mcafee_writing_cma_d
  condition: (proc.name=macompatsvc and fd.directory=/etc/cma.d)

- macro: avinetworks_supervisor_writing_ssh
  condition: >
    (proc.cmdline="se_supervisor.p /opt/avi/scripts/se_supervisor.py -d" and
      (fd.name startswith /etc/ssh/known_host_ or
       fd.name startswith /etc/ssh/ssh_monitor_config_ or
       fd.name startswith /etc/ssh/ssh_config_))
- macro: multipath_writing_conf
  condition: (proc.name = multipath and fd.name startswith /etc/multipath/)
```

このマクロに条件を追加して (おそらく別ファイルで、このマクロを上書きして)、特定のディレクトリ以下に書き込むプログラムの特定の組み合わせを可能にします。
fluentd_writing_conf_files は、書き込みを行うプログラムと変更を許可されている特定のファイルの両方を指定しているので、従うべき良い例です。

このファイルでは、ベースマクロのプログラムの一つを取り、それを繰り返しています。

```
- macro: user_known_write_etc_conditions
  condition: proc.name=confd
```

Falcoルール -3

11. etc以下への書き込み：Write below etc

etc 以下のファイルへの書き込みを試みる。

これは、ユーザーが以下のような書き込みルールのホワイトリストを拡張するためのプレースホルダです。

```
- macro: user_known_write_below_etc_activities
  condition: (never_true)

- macro: write_etc_common
  condition: >
    etc_dir and evt.dir = < and open_write
    and proc_name_exists
    and not proc.name in (passwd_binaries, shadowutils_binaries, sysdigcloud_binaries,
                          package_mgmt_binaries, ssl_mgmt_binaries, dhcp_binaries,
                          dev_creation_binaries, shell_mgmt_binaries,
                          mail_config_binaries,
                          sshkit_script_binaries,
                          ldconfig.real, ldconfig, confd, gpg, insserv,
                          apparmor_parser, update-mime, tzdata.config, tzdata.postinst,
                          systemd, systemd-machine, systemd-sysuser,
                          debconf-show, rollerd, bind9.postinst, sv,
                          gen_resolvconf., update-ca-certi, certbot, runsv,
                          qualys-cloud-ag, locales.postins, nomachine_binaries,
                          adclient, certutil, crlutil, pam-auth-update, parallels_insta,
                          openshift-launc, update-rc.d, puppet)
    and not (container and proc.cmdline in ("cp /run/secrets/kubernetes.io/serviceaccount/ca.crt
/etc/pki/ca-trust/source/anchors/openshift-ca.crt"))
    and not proc.pname in (sysdigcloud_binaries, mail_config_binaries, hddtemp.postins, sshkit_script_binaries,
locales.postins, deb_binaries, dhcp_binaries)
    and not fd.name pmatch (safe_etc_dirs)
    and not fd.name in (/etc/container_environment.sh, /etc/container_environment.json, /etc/motd,
/etc/motd.svc)
    and not sed_temporary_file
    and not exe_running_docker_save
    and not ansible_running_python
    and not python_running_denyhosts
    and not fluentd_writing_conf_files
    and not user_known_write_etc_conditions
    and not run_by_centrify
    and not run_by_adclient
    and not qualys_writing_conf_files
    and not git_writing_nssdb
    and not plesk_writing_keys
    and not plesk_install_writing_apache_conf
    and not plesk_running_mktemp
    and not networkmanager_writing_resolv_conf
    and not run_by_chef
```

```
and not add_shell_writing_shells_tmp
and not duply_writing_exclude_files
and not xmlcatalog_writing_files
and not parent_supervise_running_multilog
and not supervise_writing_status
and not pki_realm_writing_realms
and not htpasswd_writing_passwd
and not lvprogs_writing_conf
and not ovsdb_writing_openvswitch
and not datadog_writing_conf
and not curl_writing_pki_db
and not haproxy_writing_conf
and not java_writing_conf
and not dpkg_scripting
and not parent_ucf_writing_conf
and not rabbitmq_writing_conf
and not rook_writing_conf
and not php_handlers_writing_conf
and not sed_writing_temp_file
and not cron_start_writing_pam_env
and not httpd_writing_conf_logs
and not mysql_writing_conf
and not openvpn_writing_conf
and not consul_template_writing_conf
and not countly_writing_nginx_conf
and not ms_oms_writing_conf
and not ms_scx_writing_conf
and not azure_scripts_writing_conf
and not azure_networkwatcher_writing_conf
and not couchdb_writing_conf
and not update_texmf_writing_conf
and not slapadd_writing_conf
and not symantec_writing_conf
and not liveupdate_writing_conf
and not sosreport_writing_files
and not selinux_writing_conf
and not veritas_writing_config
and not nginx_writing_conf
and not nginx_writing_certs
and not chef_client_writing_conf
and not centrify_writing_krb
and not cockpit_writing_conf
and not ipsec_writing_conf
and not httpd_writing_ssl_conf
and not userhelper_writing_etc_security
and not pkgmgmt_progs_writing_pki
and not update_ca_trust_writing_pki
and not brandbot_writing_os_release
and not redis_writing_conf
and not openldap_writing_conf
and not ucpagent_writing_conf
and not iscsi_writing_conf
and not istio_writing_conf
and not ufw_writing_conf
and not calico_writing_conf
```

```
    and not calico_writing_envvars
    and not prometheus_conf_writing_conf
    and not openshift_writing_conf
    and not keepalived_writing_conf
    and not rancher_writing_conf
    and not checkpoint_writing_state
    and not jboss_in_container_writing_passwd
    and not etcd_manager_updating_dns
    and not user_known_write_below_etc_activities
    and not automount_using_mtab
    and not mcafee_writing_cma_d
    and not avinetworks_supervisor_writing_ssh
    and not multipath_writing_conf
- rule: Write below etc
  desc: an attempt to write to any file below /etc
  condition: write_etc_common
  output: "File below /etc opened for writing (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline parent=%proc.pname pcmdline=%proc.pcmdline file=%fd.name program=%proc.name
gparent=%proc.aname[2] ggparent=%proc.aname[3] gggparent=%proc.aname[4] container_id=%container.id
image=%container.image.repository)"
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

12. root以下に書き込み：Write below root

/ または /root 直下のファイルへの書き込みを試みる

```
- list: known_root_files
  items: [/root/.monit.state, /root/.auth_tokens, /root/.bash_history, /root/.ash_history,
/root/.aws/credentials,
          /root/.viminfo.tmp, /root/.lesshst, /root/.bzr.log, /root/.gitconfig.lock, /root/.babel.json,
/root/.localstack,
          /root/.node_repl_history, /root/.mongorc.js, /root/.dbshell, /root/.augeas/history, /root/.rnd,
/root/.wget-hsts, /health, /exec.fifo]

- list: known_root_directories
  items: [/root/.oracle_jre_usage, /root/.ssh, /root/.subversion, /root/.nami]

- macro: known_root_conditions
  condition: (fd.name startswith /root/orcexec.
              or fd.name startswith /root/.m2
              or fd.name startswith /root/.npm
              or fd.name startswith /root/.pki
              or fd.name startswith /root/.ivy2
              or fd.name startswith /root/.config/Cypress
              or fd.name startswith /root/.config/pulse
              or fd.name startswith /root/.config/configstore
              or fd.name startswith /root/jenkins/workspace
              or fd.name startswith /root/.jenkins
              or fd.name startswith /root/.cache
              or fd.name startswith /root/.sbt
              or fd.name startswith /root/.java
```

```
            or fd.name startswith /root/.glide
            or fd.name startswith /root/.sonar
            or fd.name startswith /root/.v8flag
            or fd.name startswith /root/infaagent
            or fd.name startswith /root/.local/lib/python
            or fd.name startswith /root/.pm2
            or fd.name startswith /root/.gnupg
            or fd.name startswith /root/.pgpass
            or fd.name startswith /root/.theano
            or fd.name startswith /root/.gradle
            or fd.name startswith /root/.android
            or fd.name startswith /root/.ansible
            or fd.name startswith /root/.crashlytics
            or fd.name startswith /root/.dbus
            or fd.name startswith /root/.composer
            or fd.name startswith /root/.gconf
            or fd.name startswith /root/.nv
            or fd.name startswith /root/.local/share/jupyter
            or fd.name startswith /root/oradiag_root
            or fd.name startswith /root/workspace
            or fd.name startswith /root/jvm
            or fd.name startswith /root/.node-gyp)

# このマクロに条件を追加して（おそらく別のファイルで、このマクロを上書きして)、
# /または/root以下の特定のディレクトリ以下に書き込むプログラムの特定の組み合わせを可能にします。

# このファイルでは、基本マクロの条件のうちの1つを取り、それを繰り返すだけです。

- macro: user_known_write_root_conditions
  condition: fd.name=/root/.bash_history

# これは、ユーザーがルートルール以下の書き込みのホワイトリストを拡張するためのプレースホルダです。
- macro: user_known_write_below_root_activities
  condition: (never_true)

- macro: runc_writing_exec_fifo
  condition: (proc.cmdline="runc:[1:CHILD] init" and fd.name=/exec.fifo)

- macro: runc_writing_var_lib_docker
  condition: (proc.cmdline="runc:[1:CHILD] init" and evt.arg.filename startswith /var/lib/docker)

- macro: mysqlsh_writing_state
  condition: (proc.name=mysqlsh and fd.directory=/root/.mysqlsh)

- rule: Write below root
  desc: an attempt to write to any file directly below / or /root
  condition: >
    root_dir and evt.dir = < and open_write
    and proc_name_exists
    and not fd.name in (known_root_files)
    and not fd.directory pmatch (known_root_directories)
    and not exe_running_docker_save
    and not gugent_writing_guestagent_log
    and not dse_writing_tmp
    and not zap_writing_state
```

```
    and not airflow_writing_state
    and not rpm_writing_root_rpmdb
    and not maven_writing_groovy
    and not chef_writing_conf
    and not kubectl_writing_state
    and not cassandra_writing_state
    and not galley_writing_state
    and not calico_writing_state
    and not rancher_writing_root
    and not runc_writing_exec_fifo
    and not mysqlsh_writing_state
    and not known_root_conditions
    and not user_known_write_root_conditions
    and not user_known_write_below_root_activities
  output: "File below / or /root opened for writing (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline parent=%proc.pname file=%fd.name program=%proc.name container_id=%container.id
image=%container.image.repository)"
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

13. 信頼されていない機密ファイルの読み取り：Read sensitive file untrusted

機密性の高いファイル(ユーザ/パスワード/認証情報を含むファイルなど)を読み込もうとした場合に検知。既知の信頼されたプログラムの場合は例外となります

```
- macro: cmp_cp_by_passwd
  condition: proc.name in (cmp, cp) and proc.pname in (passwd, run-parts)

- macro: user_known_read_sensitive_files_activities
  condition: (never_true)

- rule: Read sensitive file trusted after startup
  desc: >
    an attempt to read any sensitive file (e.g. files containing user/password/authentication
    information) by a trusted program after startup. Trusted programs might read these files
    at startup to load initial state, but not afterwards.
  condition: sensitive_files and open_read and server_procs and not proc_is_new and proc.name!="sshd" and not
user_known_read_sensitive_files_activities
  output: >
    Sensitive file opened for reading by trusted program after startup (user=%user.name
user_loginuid=%user.loginuid
    command=%proc.cmdline parent=%proc.pname file=%fd.name parent=%proc.pname gparent=%proc.aname[2]
container_id=%container.id image=%container.image.repository)
  priority: WARNING
  tags: [filesystem, mitre_credential_access]

- list: read_sensitive_file_binaries
  items: [
    iptables, ps, lsb_release, check-new-relea, dumpe2fs, accounts-daemon, sshd,
    vsftpd, systemd, mysql_install_d, psql, screen, debconf-show, sa-update,
    pam-auth-update, pam-config, /usr/sbin/spamd, polkit-agent-he, lsattr, file, sosreport,
    scxcimservera, adclient, rtvscand, cockpit-session, userhelper, ossec-syscheckd
    ]
```

```
# このマクロに条件を追加して（おそらく別ファイルで、このマクロを上書きして)、
# 特定のプログラムの組み合わせで機密ファイルにアクセスできるようにします。

# fluentd_writing_conf_files は、書き込みを行うプログラムと変更を許可される
# 特定のファイルの両方を指定しているので、それに従うのが良い例です。

# このファイルでは、基本ルールのマクロの一つを取ってそれを繰り返しています。

- macro: user_read_sensitive_file_conditions
  condition: cmp_cp_by_passwd

- list: read_sensitive_file_images
  items: []

- macro: user_read_sensitive_file_containers
  condition: (container and container.image.repository in (read_sensitive_file_images))

- rule: Read sensitive file untrusted
  desc: >
    an attempt to read any sensitive file (e.g. files containing user/password/authentication
    information). Exceptions are made for known trusted programs.
  condition: >
    sensitive_files and open_read
    and proc_name_exists
    and not proc.name in (user_mgmt_binaries, userexec_binaries, package_mgmt_binaries,
     cron_binaries, read_sensitive_file_binaries, shell_binaries, hids_binaries,
     vpn_binaries, mail_config_binaries, nomachine_binaries, sshkit_script_binaries,
     in.proftpd, mandb, salt-minion, postgres_mgmt_binaries,
     google_oslogin_
     )
    and not cmp_cp_by_passwd
    and not ansible_running_python
    and not proc.cmdline contains /usr/bin/mandb
    and not run_by_qualys
    and not run_by_chef
    and not run_by_google_accounts_daemon
    and not user_read_sensitive_file_conditions
    and not perl_running_plesk
    and not perl_running_updmap
    and not veritas_driver_script
    and not perl_running_centrifydc
    and not runuser_reading_pam
    and not linux_bench_reading_etc_shadow
    and not user_known_read_sensitive_files_activities
    and not user_read_sensitive_file_containers
  output: >
    Sensitive file opened for reading by non-trusted program (user=%user.name user_loginuid=%user.loginuid
program=%proc.name
    command=%proc.cmdline file=%fd.name parent=%proc.pname gparent=%proc.aname[2] ggparent=%proc.aname[3]
gggparent=%proc.aname[4] container_id=%container.id image=%container.image.repository)
  priority: WARNING
  tags: [filesystem, mitre_credential_access, mitre_discovery]
```

14. rpmデータベース以下への書き込み：Write below rpm database

rpm 以外の関連プログラムで rpm データベースへの書き込みを試みた場合に検知

```
- macro: amazon_linux_running_python_yum
  condition: >
    (proc.name = python and
     proc.pcmdline = "python -m amazon_linux_extras system_motd" and
     proc.cmdline startswith "python -c import yum;")
- macro: user_known_write_rpm_database_activities
  condition: (never_true)

# Only let rpm-related programs write to the rpm database
- rule: Write below rpm database
  desc: an attempt to write to the rpm database by any non-rpm related program
  condition: >
    fd.name startswith /var/lib/rpm and open_write
    and not rpm_procs
    and not ansible_running_python
    and not python_running_chef
    and not exe_running_docker_save
    and not amazon_linux_running_python_yum
    and not user_known_write_rpm_database_activities
  output: "Rpm database opened for writing by a non-rpm program (command=%proc.cmdline file=%fd.name
parent=%proc.pname pcmdline=%proc.pcmdline container_id=%container.id image=%container.image.repository)"
  priority: ERROR
  tags: [filesystem, software_mgmt, mitre_persistence]
```

15. DBプログラムがプロセスを生成：DB program spawned process

これは、データベースサーバ関連のプログラムが自分自身とは別の新しいプロセスを生成した場合に発生してはならないものであり、いくつかのSQLインジェクション攻撃の後に続くものです。

```
- macro: postgres_running_wal_e
  condition: (proc.pname=postgres and proc.cmdline startswith "sh -c envdir /etc/wal-e.d/env
/usr/local/bin/wal-e")

- macro: redis_running_prepost_scripts
  condition: (proc.aname[2]=redis-server and (proc.cmdline contains "redis-server.post-up.d" or proc.cmdline
contains "redis-server.pre-up.d"))

- macro: rabbitmq_running_scripts
  condition: >
    (proc.pname=beam.smp and
    (proc.cmdline startswith "sh -c exec ps" or
     proc.cmdline startswith "sh -c exec inet_gethost" or
     proc.cmdline= "sh -s unix:cmd" or
     proc.cmdline= "sh -c exec /bin/sh -s unix:cmd 2>&1"))
- macro: rabbitmqctl_running_scripts
  condition: (proc.aname[2]=rabbitmqctl and proc.cmdline startswith "sh -c ")
```

```
- macro: run_by_appdynamics
  condition: (proc.pname=java and proc.pcmdline startswith "java -jar -Dappdynamics")

- macro: user_known_db_spawned_processes
  condition: (never_true)

- rule: DB program spawned process
  desc: >
    a database-server related program spawned a new process other than itself.
    This shouldn\'t occur and is a follow on from some SQL injection attacks.
  condition: >
    proc.pname in (db_server_binaries)
    and spawned_process
    and not proc.name in (db_server_binaries)
    and not postgres_running_wal_e
    and not user_known_db_spawned_processes
  output: >
    Database-related program spawned process other than itself (user=%user.name user_loginuid=%user.loginuid
    program=%proc.cmdline parent=%proc.pname container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [process, database, mitre_execution]
```

16. バイナリーディレクトリの更新：Modify binary dirs

バイナリディレクトリのセット以下のファイルを変更しようとする試み。

```
- macro: user_known_modify_bin_dir_activities
  condition: (never_true)

- rule: Modify binary dirs
  desc: an attempt to modify any file below a set of binary directories.
  condition: bin_dir_rename and modify and not package_mgmt_procs and not exe_running_docker_save and not
user_known_modify_bin_dir_activities
  output: >
    File below known binary directory renamed/removed (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline
    pcmdline=%proc.pcmdline operation=%evt.type file=%fd.name %evt.args container_id=%container.id
image=%container.image.repository)
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

17. バイナリーディレクトリー作成：Mkdir binary dirs

バイナリーディレクトリのセット下にディレクトリを作成しようとする試み。

```
- macro: user_known_mkdir_bin_dir_activities
  condition: (never_true)

- rule: Mkdir binary dirs
  desc: an attempt to create a directory below a set of binary directories.
```

```
  condition: mkdir and bin_dir_mkdir and not package_mgmt_procs and not user_known_mkdir_bin_dir_activities
  output: >
    Directory below known binary directory created (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline directory=%evt.arg.path container_id=%container.id image=%container.image.repository)
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

18. thread namespaceの変更：Change thread namespace

setns を呼び出すことで、プログラム/スレッドのネームスペースを変更しようとする試み (一般的にはコンテナの作成の一部として行われます)。

```
# このリストでは、スレッドネームスペースの変更ルール全体をコピーして上書きすることなく、
# スレッドネームスペースの変更を許可するコマンドのセットを簡単に追加することができます。
- list: user_known_change_thread_namespace_binaries
  items: []

- macro: user_known_change_thread_namespace_activities
  condition: (never_true)

- list: network_plugin_binaries
  items: [aws-cni, azure-vnet]

- macro: calico_node
  condition: (container.image.repository endswith calico/node and proc.name=calico-node)

- macro: weaveworks_scope
  condition: (container.image.repository endswith weaveworks/scope and proc.name=scope)

- rule: Change thread namespace
  desc: >
    an attempt to change a program/thread\'s namespace (commonly done
    as a part of creating a container) by calling setns.
  condition: >
    evt.type=setns and evt.dir=<
    and proc_name_exists
    and not (container.id=host and proc.name in (docker_binaries, k8s_binaries, lxd_binaries, nsenter))
    and not proc.name in (sysdigcloud_binaries, sysdig, calico, oci-umount, cilium-cni, network_plugin_binaries)
    and not proc.name in (user_known_change_thread_namespace_binaries)
    and not proc.name startswith "runc"
    and not proc.cmdline startswith "containerd"
    and not proc.pname in (sysdigcloud_binaries, hyperkube, kubelet, protokube, dockerd, tini, aws)
    and not python_running_sdchecks
    and not java_running_sdjagent
    and not kubelet_running_loopback
    and not rancher_agent
    and not rancher_network_manager
    and not calico_node
    and not weaveworks_scope
    and not user_known_change_thread_namespace_activities
  output: >
    Namespace change (setns) by unexpected program (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline
```

```
    parent=%proc.pname %container.info container_id=%container.id
image=%container.image.repository:%container.image.tag)
  priority: NOTICE
  tags: [process, mitre_privilege_escalation, mitre_lateral_movement]
```

19. 信頼されていないシェルを実行する：Run shell untrusted

シェルではないアプリケーションの下にシェルを生成しようとする試み。特定のアプリケーションを監視します。

このリストにあるバイナリとその派生は、シェルを生成することができません。これには、間接的にだけでなく、直接シェルを生成するバイナリも含まれます。例えば、mod_{php,perl} のための apache -> php/perl -> いくつかのシェルも許可されていません。

```
- list: protected_shell_spawning_binaries
  items: [
    http_server_binaries, db_server_binaries, nosql_server_binaries, mail_binaries,
    fluentd, flanneld, splunkd, consul, smbd, runsv, PM2
    ]

- macro: parent_java_running_zookeeper
  condition: (proc.pname=java and proc.pcmdline contains org.apache.zookeeper.server)

- macro: parent_java_running_kafka
  condition: (proc.pname=java and proc.pcmdline contains kafka.Kafka)

- macro: parent_java_running_elasticsearch
  condition: (proc.pname=java and proc.pcmdline contains org.elasticsearch.bootstrap.Elasticsearch)

- macro: parent_java_running_activemq
  condition: (proc.pname=java and proc.pcmdline contains activemq.jar)

- macro: parent_java_running_cassandra
  condition: (proc.pname=java and (proc.pcmdline contains "-Dcassandra.config.loader" or proc.pcmdline contains
org.apache.cassandra.service.CassandraDaemon))

- macro: parent_java_running_jboss_wildfly
  condition: (proc.pname=java and proc.pcmdline contains org.jboss)

- macro: parent_java_running_glassfish
  condition: (proc.pname=java and proc.pcmdline contains com.sun.enterprise.glassfish)

- macro: parent_java_running_hadoop
  condition: (proc.pname=java and proc.pcmdline contains org.apache.hadoop)

- macro: parent_java_running_datastax
  condition: (proc.pname=java and proc.pcmdline contains com.datastax)

- macro: nginx_starting_nginx
  condition: (proc.pname=nginx and proc.cmdline contains "/usr/sbin/nginx -c /etc/nginx/nginx.conf")
```

```
- macro: nginx_running_aws_s3_cp
  condition: (proc.pname=nginx and proc.cmdline startswith "sh -c /usr/local/bin/aws s3 cp")

- macro: consul_running_net_scripts
  condition: (proc.pname=consul and (proc.cmdline startswith "sh -c curl" or proc.cmdline startswith "sh -c
nc"))

- macro: consul_running_alert_checks
  condition: (proc.pname=consul and proc.cmdline startswith "sh -c /bin/consul-alerts")

- macro: serf_script
  condition: (proc.cmdline startswith "sh -c serf")

- macro: check_process_status
  condition: (proc.cmdline startswith "sh -c kill -0 ")
```

場合によっては、コンテナ内で直接実行されているノードプロセスを保護されたシェルスポーンナとして考慮するとよいでしょう。例えば、コンテナの直接の入り口として pm2-docker や pm2 start some-app.js --no-daemon-mode を使用したり、ノードアプリが express のようなものを使用して長寿命のサーバである場合などです。

しかし、パイプラインの構築に関連したnodeの他の用途もあり、その場合、nodeはサーバーではなく一般的なスクリプトツールです。このような場合、シェルを使用する可能性が高く、このような場合には、ノードプロセスをシェルスポーンナとして保護することは考えたくありません。

これらのケースのいずれかを選択しなければならないので、デフォルトではノードプロセスは保護されていないと考えています。コンテナ内で実行されているノードプロセスを保護されたシェルスポーンナとみなしたい場合は、以下のマクロをオーバーライドして "never_true "節を削除することで有効になります。

```
- macro: possibly_node_in_container
  condition: (never_true and (proc.pname=node and proc.aname[3]=docker-containe))
```

同様に、apache tomcat によってスポーンされたシェルはすべて疑わしいと考えた方がよいでしょう。有名な apache struts 攻撃 (CVE-2017-5638) は、シェルをスポーンするようなことをするために悪用される可能性があります。

しかし、多くのアプリケーションでは、ビルドパイプラインの一部として、任意のシェルを実行するために tomcat を使用しています。

node の場合と同様に、この場合も opt-in としています。

```
- macro: possibly_parent_java_running_tomcat
  condition: (never_true and proc.pname=java and proc.pcmdline contains org.apache.catalina.startup.Bootstrap)

- macro: protected_shell_spawner
  condition: >
    (proc.aname in (protected_shell_spawning_binaries)
    or parent_java_running_zookeeper
    or parent_java_running_kafka
    or parent_java_running_elasticsearch
    or parent_java_running_activemq
    or parent_java_running_cassandra
    or parent_java_running_jboss_wildfly
    or parent_java_running_glassfish
    or parent_java_running_hadoop
    or parent_java_running_datastax
    or possibly_parent_java_running_tomcat
    or possibly_node_in_container)
- list: mesos_shell_binaries
  items: [mesos-docker-ex, mesos-slave, mesos-health-ch]
```

runsvはprotected_shell_spawnerとpnameによる排他の両方にあることに注意してください。これは、runv自身はシェルをスポーンすることができますが（./runスクリプトと./finishスクリプト）、runvプロセスはシェルをスポーンすることができないことを意味します。

```
- rule: Run shell untrusted
  desc: an attempt to spawn a shell below a non-shell application. Specific applications are monitored.
  condition: >
    spawned_process
    and shell_procs
    and proc.pname exists
    and protected_shell_spawner
    and not proc.pname in (shell_binaries, gitlab_binaries, cron_binaries, user_known_shell_spawn_binaries,
                           needrestart_binaries,
                           mesos_shell_binaries,
                           erl_child_setup, exechealthz,
                           PM2, PassengerWatchd, c_rehash, svlogd, logrotate, hhvm, serf,
                           lb-controller, nvidia-installe, runsv, statsite, erlexec, calico-node,
                           "puma reactor")
    and not proc.cmdline in (known_shell_spawn_cmdlines)
    and not proc.aname in (unicorn_launche)
    and not consul_running_net_scripts
    and not consul_running_alert_checks
    and not nginx_starting_nginx
    and not nginx_running_aws_s3_cp
    and not run_by_package_mgmt_binaries
    and not serf_script
    and not check_process_status
    and not run_by_foreman
    and not python_mesos_marathon_scripting
```

```
    and not splunk_running_forwarder
    and not postgres_running_wal_e
    and not redis_running_prepost_scripts
    and not rabbitmq_running_scripts
    and not rabbitmqctl_running_scripts
    and not run_by_appdynamics
    and not user_shell_container_exclusions
  output: >
    Shell spawned by untrusted binary (user=%user.name user_loginuid=%user.loginuid shell=%proc.name
parent=%proc.pname
    cmdline=%proc.cmdline pcmdline=%proc.pcmdline gparent=%proc.aname[2] ggparent=%proc.aname[3]
    aname[4]=%proc.aname[4] aname[5]=%proc.aname[5] aname[6]=%proc.aname[6] aname[7]=%proc.aname[7]
container_id=%container.id image=%container.image.repository)
  priority: DEBUG
  tags: [shell, mitre_execution]
```

共通マクロ、リスト - 4

```
- macro: allowed_openshift_registry_root
  condition: >
    (container.image.repository startswith openshift3/ or
     container.image.repository startswith registry.redhat.io/openshift3/ or
     container.image.repository startswith registry.access.redhat.com/openshift3/)
# Source: https://docs.openshift.com/enterprise/3.2/install_config/install/disconnected_install.html
- macro: openshift_image
  condition: >
    (allowed_openshift_registry_root and
      (container.image.repository endswith /logging-deployment or
       container.image.repository endswith /logging-elasticsearch or
       container.image.repository endswith /logging-kibana or
       container.image.repository endswith /logging-fluentd or
       container.image.repository endswith /logging-auth-proxy or
       container.image.repository endswith /metrics-deployer or
       container.image.repository endswith /metrics-hawkular-metrics or
       container.image.repository endswith /metrics-cassandra or
       container.image.repository endswith /metrics-heapster or
       container.image.repository endswith /ose-haproxy-router or
       container.image.repository endswith /ose-deployer or
       container.image.repository endswith /ose-sti-builder or
       container.image.repository endswith /ose-docker-builder or
       container.image.repository endswith /ose-pod or
       container.image.repository endswith /ose-node or
       container.image.repository endswith /ose-docker-registry or
       container.image.repository endswith /prometheus-node-exporter or
       container.image.repository endswith /image-inspector))
```

これらのイメージは --privileged での実行と、ホストファイルシステムからの機密パスのマウントの両方が許可されています。

注意: このリストは、古いローカルの falco ルールファイルで trusted_images に追加されている可能性のあるファイルとの下位互換性のためだけに提供されています。カスタマイズするには、privileged_images または falco_sensitive_mount_images のいずれかにイメージを追加した方が良いでしょう。

```
- list: trusted_images
  items: []
```

注意: このマクロは、trusted_imagesにアペンディングしている可能性のある古いローカルfalcoルールファイルとの下位互換性のためだけに提供されています。カスタマイズするには、user_trusted_containers、user_privileged_containers、user_sensitive_mount_containers にコンテナを追加した方が良いでしょう。

```
- macro: trusted_containers
  condition: (container.image.repository in (trusted_images))
```

このマクロに条件を追加して (おそらく別のファイルで、このマクロを上書きして)、信頼されていて、それゆえに機密性の高いマウントでの特権的な *and* 実行を許可されている追加のコンテナを指定するようにします。

trusted_images と同様に、これは user_privileged_containers と user_sensitive_mount_containers を優先して非推奨とされており、下位互換性のためにのみ提供されています。

このファイルでは、trusteded_containers にあるイメージのうちのひとつを取り出してそれを繰り返すだけです。

```
- macro: user_trusted_containers
  condition: (never_true)

- list: sematext_images
  items: [docker.io/sematext/sematext-agent-docker, docker.io/sematext/agent, docker.io/sematext/logagent,
          registry.access.redhat.com/sematext/sematext-agent-docker,
          registry.access.redhat.com/sematext/agent,
          registry.access.redhat.com/sematext/logagent]

# These container images are allowed to run with --privileged
- list: falco_privileged_images
  items: [
    docker.io/calico/node,
    calico/node,
    docker.io/cloudnativelabs/kube-router,
    docker.io/docker/ucp-agent,
    docker.io/falcosecurity/falco,
```

```
    docker.io/mesosphere/mesos-slave,
    docker.io/rook/toolbox,
    docker.io/sysdig/falco,
    docker.io/sysdig/sysdig,
    falcosecurity/falco,
    gcr.io/google_containers/kube-proxy,
    gcr.io/google-containers/startup-script,
    gcr.io/projectcalico-org/node,
    gke.gcr.io/kube-proxy,
    gke.gcr.io/gke-metadata-server,
    gke.gcr.io/netd-amd64,
    gcr.io/google-containers/prometheus-to-sd,
    k8s.gcr.io/ip-masq-agent-amd64,
    k8s.gcr.io/kube-proxy,
    k8s.gcr.io/prometheus-to-sd,
    quay.io/calico/node,
    sysdig/falco,
    sysdig/sysdig,
    sematext_images
    ]

- macro: falco_privileged_containers
  condition: (openshift_image or
              user_trusted_containers or
              container.image.repository in (trusted_images) or
              container.image.repository in (falco_privileged_images) or
              container.image.repository startswith istio/proxy_ or
              container.image.repository startswith quay.io/sysdig/)
```

このマクロに条件を追加して (おそらく別のファイルで、このマクロを上書きして)、特権実行を許可される追加のコンテナを指定します。

このファイルでは、falco_privileged_imagesにあるイメージのうちの1つを取り、それを繰り返すだけです。

```
- macro: user_privileged_containers
  condition: (never_true)

- list: rancher_images
  items: [
    rancher/network-manager, rancher/dns, rancher/agent,
    rancher/lb-service-haproxy, rancher/metadata, rancher/healthcheck
  ]
```

これらのコンテナイメージは、ホストファイルシステムからの機密パスのマウントを許可されています。

```
- list: falco_sensitive_mount_images
  items: [
```

```
    docker.io/sysdig/falco, docker.io/sysdig/sysdig, sysdig/falco, sysdig/sysdig,
    docker.io/falcosecurity/falco, falcosecurity/falco,
    gcr.io/google_containers/hyperkube,
    gcr.io/google_containers/kube-proxy, docker.io/calico/node,
    docker.io/rook/toolbox, docker.io/cloudnativelabs/kube-router, docker.io/consul,
    docker.io/datadog/docker-dd-agent, docker.io/datadog/agent, docker.io/docker/ucp-agent,
 docker.io/gliderlabs/logspout,
    docker.io/netdata/netdata, docker.io/google/cadvisor, docker.io/prom/node-exporter,
    amazon/amazon-ecs-agent
    ]

 - macro: falco_sensitive_mount_containers
   condition: (user_trusted_containers or
               container.image.repository in (trusted_images) or
               container.image.repository in (falco_sensitive_mount_images) or
               container.image.repository startswith quay.io/sysdig/)
```

これらのコンテナイメージは hostnetwork=true での実行が許可されています。

```
- list: falco_hostnetwork_images
  items: [
    gcr.io/google-containers/prometheus-to-sd,
    gcr.io/projectcalico-org/typha,
    gcr.io/projectcalico-org/node,
    gke.gcr.io/gke-metadata-server,
    gke.gcr.io/kube-proxy,
    gke.gcr.io/netd-amd64,
    k8s.gcr.io/ip-masq-agent-amd64
    k8s.gcr.io/prometheus-to-sd,
    ]
```

Falcoルール - 4

20. 特権コンテナの起動：Launch Privileged Container

特権コンテナで起動した初期プロセスを検出します。既知の信頼されたイメージについては例外とします。

このマクロに条件を追加して (おそらく別ファイルで、このマクロを上書きして)、センシティブなマウントを実行できる追加のコンテナを指定します。このファイルでは、falco_sensitive_mount_imagesにあるイメージのうちの1つを取り、それを繰り返すだけです。

```
 - macro: user_sensitive_mount_containers
   condition: (never_true)

 - rule: Launch Privileged Container
```

```
  desc: Detect the initial process started in a privileged container. Exceptions are made for known trusted
images.
  condition: >
    container_started and container
    and container.privileged=true
    and not falco_privileged_containers
    and not user_privileged_containers
  output: Privileged container started (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
%container.info image=%container.image.repository:%container.image.tag)
  priority: INFO
  tags: [container, cis, mitre_privilege_escalation, mitre_lateral_movement]
```

21. 機密マウントコンテナを起動する：Launch Sensitive Mount Container

機密のホストディレクトリからマウントされているコンテナが起動した初期プロセスを検出します。今のところ、/etc のフルマウントのみを機密として考慮しています。理想的には /etc 以下のすべてのサブディレクトリも同様に考慮することになりますが、sysdig が使用しているglobの仕組みはフルパターンの除外を許可しておらず、1文字のみを除外しています。

```
- macro: sensitive_mount
  condition: (container.mount.dest[/proc*] != "N/A" or
              container.mount.dest[/var/run/docker.sock] != "N/A" or
              container.mount.dest[/var/run/crio/crio.sock] != "N/A" or
              container.mount.dest[/var/lib/kubelet] != "N/A" or
              container.mount.dest[/var/lib/kubelet/pki] != "N/A" or
              container.mount.dest[/] != "N/A" or
              container.mount.dest[/home/admin] != "N/A" or
              container.mount.dest[/etc] != "N/A" or
              container.mount.dest[/etc/kubernetes] != "N/A" or
              container.mount.dest[/etc/kubernetes/manifests] != "N/A" or
              container.mount.dest[/root*] != "N/A")
```

libcontainer がコンテナのルートプログラムをセットアップするために実行するステップは以下の通りです。

- clone + exec self to a program runc:[0:PARENT]
- clone a program runc:[1:CHILD] which sets up all the namespaces
- clone a second program runc:[2:INIT] + exec to the root program.

runc:[2:INIT]のparentは、runc:[0:PARENT]です。
1:CHILDが作成されるとすぐに0:PARENTが終了するので、2:INITがルートプログラムを実行した時点で0:PARENTはすでに終了しているか、まだその辺にいるかもしれないという競争になります。そのため, 両方を処理します.

また、runc:[1:CHILD]を親プロセスとしてカウントするようにしていますが、これはイベントを失ったり、状態を把握できなくなったりしたときに発生する可能性があります。

```
- macro: container_entrypoint
  condition: (not proc.pname exists or proc.pname in (runc:[0:PARENT], runc:[1:CHILD], runc, docker-runc, exe,
docker-runc-cur))

- rule: Launch Sensitive Mount Container
  desc: >
    Detect the initial process started by a container that has a mount from a sensitive host directory
    (i.e. /proc). Exceptions are made for known trusted images.
  condition: >
    container_started and container
    and sensitive_mount
    and not falco_sensitive_mount_containers
    and not user_sensitive_mount_containers
  output: Container with sensitive mount started (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline %container.info image=%container.image.repository:%container.image.tag
mounts=%container.mounts)
  priority: INFO
  tags: [container, cis, mitre_lateral_movement]
```

22. 起動禁止コンテナ：Launch Disallowed Container

許可されているコンテナのリストにないコンテナによって開始された初期プロセスを検出します。

ローカル/ユーザールールファイルでは、このマクロをオーバーライドして、環境で実行したいコンテナイメージを明示的に列挙することができます。このメインの falco ルールファイルでは、実行可能なすべてのコンテナを知る方法はありませんので、真に評価されることが保証されているフィルタを使用して、どのコンテナも許可されます。オーバーライドマクロでは、条件は次のようになります。

(container.image.repository = vendor/container-1 または
container.image.repository = vendor/container-2 or ....)

```
- macro: allowed_containers
  condition: (container.id exists)

- rule: Launch Disallowed Container
  desc: >
    Detect the initial process started by a container that is not in a list of allowed containers.
  condition: container_started and container and not allowed_containers
  output: Container started and not in allowed list (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline %container.info image=%container.image.repository:%container.image.tag)
  priority: WARNING
  tags: [container, mitre_lateral_movement]
```

23. インタラクティブなシステムユーザ：System user interactive

システム (つまりログインしていない) ユーザが対話型コマンドを実行しようとすること。

```
- macro: user_known_system_user_login
  condition: (never_true)

# ルートによって対話的に実行されるもの
# - condition: evt.type != switch and user.name = root and proc.name != sshd and interactive
#  output: "Interactive root (%user.name %proc.name %evt.dir %evt.type %evt.args %fd.name)"
#  priority: WARNING

- rule: System user interactive
  desc: an attempt to run interactive commands by a system (i.e. non-login) user
  condition: spawned_process and system_users and interactive and not user_known_system_user_login
  output: "System user ran an interactive command (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline container_id=%container.id image=%container.image.repository)"
  priority: INFO
  tags: [users, mitre_remote_access_tools]
```

24. コンテナ内のターミナルシェル：Terminal shell in container

ターミナルが付属しているコンテナへの入り口/実行ポイントとしてシェルを使用した時に検出

場合によっては、シェルはコンテナ内で実行されることが予想されます。例えば、設定管理ソフトウェアがこれを行う場合がありますが、これは想定されています。

```
- macro: user_expected_terminal_shell_in_container_conditions
  condition: (never_true)

- rule: Terminal shell in container
  desc: A shell was used as the entrypoint/exec point into a container with an attached terminal.
  condition: >
    spawned_process and container
    and shell_procs and proc.tty != 0
    and container_entrypoint
    and not user_expected_terminal_shell_in_container_conditions
  output: >
    A shell was spawned in a container with an attached terminal (user=%user.name user_loginuid=%user.loginuid
%container.info
    shell=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%container.id
image=%container.image.repository)
  priority: NOTICE
  tags: [container, shell, mitre_execution]
```

共通マクロ、リスト - 5

ある種のコンテナタイプ(mesos)では、扱うコンテナイメージがなく、コンテナ名も自動生成されるので、安定したソフトウェアとしての側面がありません。このような場合は、特定のコマンドラインを許可するようにします。

```
- list: known_shell_spawn_cmdlines
  items: [
    '"sh -c uname -p 2> /dev/null"',
    '"sh -c uname -s 2>&1"',
    '"sh -c uname -r 2>&1"',
    '"sh -c uname -v 2>&1"',
    '"sh -c uname -a 2>&1"',
    '"sh -c ruby -v 2>&1"',
    '"sh -c getconf CLK_TCK"',
    '"sh -c getconf PAGESIZE"',
    '"sh -c LC_ALL=C LANG=C /sbin/ldconfig -p 2>/dev/null"',
    '"sh -c LANG=C /sbin/ldconfig -p 2>/dev/null"',
    '"sh -c /sbin/ldconfig -p 2>/dev/null"',
    '"sh -c stty -a 2>/dev/null"',
    '"sh -c stty -a < /dev/tty"',
    '"sh -c stty -g < /dev/tty"',
    '"sh -c node index.js"',
    '"sh -c node index"',
    '"sh -c node ./src/start.js"',
    '"sh -c node app.js"',
    '"sh -c node -e \"require(''nan'')\""',
    '"sh -c node -e \"require(''nan'')\")"',
    '"sh -c node $NODE_DEBUG_OPTION index.js "',
    '"sh -c crontab -l 2"',
    '"sh -c lsb_release -a"',
    '"sh -c lsb_release -is 2>/dev/null"',
    '"sh -c whoami"',
    '"sh -c node_modules/.bin/bower-installer"',
    '"sh -c /bin/hostname -f 2> /dev/null"',
    '"sh -c locale -a"',
    '"sh -c  -t -i"',
    '"sh -c openssl version"',
    '"bash -c id -Gn kafadmin"',
    '"sh -c /bin/sh -c ''date +%%s''"'
    ]
```

このリストでは、コンテナ内でシェルを実行するために許可されているコマンドのセットを、コンテナ内のランシェルマクロ全体をコピーして上書きすることなく、簡単に追加することができます。https://github.com/draios/falco/issues/255 が修正されると、誰かが既存のリストのいずれかを追加できるようになります。

```
- list: user_known_shell_spawn_binaries
  items: []
```

このマクロは、ルール全体をオーバーライドすることなく、コンテナ内でシェルを実行するために許可されるコマンドのセットを簡単に追加することができます。デフォルト値は常に false の式で、ルールの "not ...." が適用されると真になります。

```
- macro: user_shell_container_exclusions
  condition: (never_true)
```

Falcoルール - 5

25. System procsネットワークアクティビティ：System procs network activity

ネットワークトラフィックを送受信することが予期されていないシステムバイナリによって実行されるネットワークアクティビティを検出

```
- macro: login_doing_dns_lookup
  condition: (proc.name=login and fd.l4proto=udp and fd.sport=53)

# sockfamily ip は、Unix ドメインのソケットで通信する特定のプロセス（「グループ」のようなもの）
# を除外するためのものです。
- rule: System procs network activity
  desc: any network activity performed by system binaries that are not expected to send or receive any network
traffic
  condition: >
    (fd.sockfamily = ip and (system_procs or proc.name in (shell_binaries)))
    and (inbound_outbound)
    and not proc.name in (known_system_procs_network_activity_binaries)
    and not login_doing_dns_lookup
    and not user_expected_system_procs_network_activity_conditions
  output: >
    Known system binary sent/received network traffic
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline connection=%fd.name
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_exfiltration]
```

26. 許可されていない http proxy env でプログラムを実行：Program run with disallowed http proxy env

許可されていない HTTP_PROXY 環境変数を使用してプログラムを実行した時に検出

このリストは、ネットワーク上で通信することが期待されるシステム proc 名を簡単にホワイトリスト化することができます。

```
- list: known_system_procs_network_activity_binaries
```

```
  items: [systemd, hostid, id]
```

このマクロは、システムバイナリがネットワーク上で通信を許可する条件を指定することができます。例えば、特定の proc.cmdline の値だけを許可するなど、許可される条件をより細かく指定することができます。

```
- macro: user_expected_system_procs_network_activity_conditions
  condition: (never_true)
```

記述すると以下のようになります。
(proc.env contains "HTTP_PROXY=http://my.http.proxy.com ")
最後のスペースは意図的なものなので、 実際のプロキシの接頭辞でのマッチングを避けます。

```
- macro: allowed_ssh_proxy_env
  condition: (always_true)

- list: http_proxy_binaries
  items: [curl, wget]

- macro: http_proxy_procs
  condition: (proc.name in (http_proxy_binaries))

- rule: Program run with disallowed http proxy env
  desc: An attempt to run a program with a disallowed HTTP_PROXY environment variable
  condition: >
    spawned_process and
    http_proxy_procs and
    not allowed_ssh_proxy_env and
    proc.env icontains HTTP_PROXY
  output: >
    Program run with disallowed HTTP_PROXY environment variable
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline env=%proc.env parent=%proc.pname
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [host, users]
```

27. 解釈されたprocsアウトバウンドネットワークアクティビティ：Interpreted procs outbound network activity

任意の解釈されたプログラム（perl、python、rubyなど）によって実行されるアウトバウンドネットワーク活動。

いくつかの環境では、解釈されたプログラム (perl,python,ruby など) が着信接続をリッスンしたり、発信接続を実行しようとすると、 不審に思うかもしれません。これらのルールはデフォルトでは有効になっていませんが、以下のマクロを変更して有効にすることができます。

```
- macro: consider_interpreted_inbound
  condition: (never_true)

- macro: consider_interpreted_outbound
  condition: (never_true)

- rule: Interpreted procs inbound network activity
  desc: Any inbound network activity performed by any interpreted program (perl, python, ruby, etc.)
  condition: >
    (inbound and consider_interpreted_inbound
     and interpreted_procs)
  output: >
    Interpreted program received/listened for network traffic
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline connection=%fd.name
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_exfiltration]

- rule: Interpreted procs outbound network activity
  desc: Any outbound network activity performed by any interpreted program (perl, python, ruby, etc.)
  condition: >
    (outbound and consider_interpreted_outbound
     and interpreted_procs)
  output: >
    Interpreted program performed outgoing network connection
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline connection=%fd.name
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_exfiltration]
```

28. 予期しないUDPトラフィック：Unexpected UDP Traffic

ポート53(DNS)など一般的に使用されているポートではないUDPトラフィック。

```
- list: openvpn_udp_ports
  items: [1194, 1197, 1198, 8080, 9201]

- list: l2tp_udp_ports
  items: [500, 1701, 4500, 10000]
```

```
- list: statsd_ports
  items: [8125]

- list: ntp_ports
  items: [123]
```

アプリケーションによっては、接続性をテストするためだけに udp ソケットをアドレスに接続します。udp接続が動作すると仮定して、実際にデータを送受信するtcp接続を行います。
このことを念頭に置き、いくつかの誤検知を避けるために、ここでは一般的に見られるいくつかのポートをリストアップしました。さらに、メインルールをオプトインにしているので、デフォルトでは無効になっています。

```
- list: test_connect_ports
  items: [0, 9, 80, 3306]

- macro: do_unexpected_udp_check
  condition: (never_true)

- list: expected_udp_ports
  items: [53, openvpn_udp_ports, l2tp_udp_ports, statsd_ports, ntp_ports, test_connect_ports]

- macro: expected_udp_traffic
  condition: fd.port in (expected_udp_ports)

- rule: Unexpected UDP Traffic
  desc: UDP traffic not on port 53 (DNS) or other commonly used ports
  condition: (inbound_outbound) and do_unexpected_udp_check and fd.l4proto=udp and not expected_udp_traffic
  output: >
    Unexpected UDP Traffic Seen
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline connection=%fd.name proto=%fd.l4proto
evt=%evt.type %evt.args container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, mitre_exfiltration]
```

29. sudoでないsetuid：Non sudo setuid

setuid を呼び出してユーザを変更しようとする試み。sudo/su は除外されます。 setuid の呼び出しは通常権限の削除を伴うので、"root" と "nobody" のユーザが自分自身を訴えている場合も除外されます。

現在のfalcoが処理するシステムコールの制限では(例: read/write/sendto/recvfrom/etcを除外した場合、このルールは発動しません)。

- rule: Ssh error in syslog
  desc: any ssh errors (failed logins, disconnects, ...) sent to syslog
  condition: syslog and ssh_error_message and evt.dir = <

output: "sshd sent error message to syslog (error=%evt.buffer)"
priority: WARNING

```
- macro: somebody_becoming_themself
  condition: ((user.name=nobody and evt.arg.uid=nobody) or
              (user.name=www-data and evt.arg.uid=www-data) or
              (user.name=_apt and evt.arg.uid=_apt) or
              (user.name=postfix and evt.arg.uid=postfix) or
              (user.name=pki-agent and evt.arg.uid=pki-agent) or
              (user.name=pki-acme and evt.arg.uid=pki-acme) or
              (user.name=nfsnobody and evt.arg.uid=nfsnobody) or
              (user.name=postgres and evt.arg.uid=postgres))

- macro: nrpe_becoming_nagios
  condition: (proc.name=nrpe and evt.arg.uid=nagios)
```

コンテナでは、ユーザ名はコンテナ内には存在するがホスト上には存在しない uid のためのものかもしれません。(https://github.com/draios/sysdig/issues/954 を参照)。その場合は setuid を許可してください。

```
- macro: known_user_in_container
  condition: (container and user.name != "N/A")
```

setuidを呼び出してユーザーを変更するプログラムの特定の組み合わせを可能にするために、このマクロに条件を追加します（おそらく別のファイルで、このマクロを上書きします）。このファイルでは、ベースとなるマクロの条件のうちの1つを取り、それを繰り返すだけです。

```
- macro: user_known_non_sudo_setuid_conditions
  condition: user.name=root
```

sshd の場合、メールプログラムは非 root として動作していても root に設定しようとします。無意味な FP を避けるために、ここでは除外しています。

```
- rule: Non sudo setuid
  desc: >
    an attempt to change users by calling setuid. sudo/su are excluded. users "root" and "nobody"
    suing to itself are also excluded, as setuid calls typically involve dropping privileges.
  condition: >
    evt.type=setuid and evt.dir=>
    and (known_user_in_container or not container)
    and not user.name=root
    and not somebody_becoming_themself
    and not proc.name in (known_setuid_binaries, userexec_binaries, mail_binaries, docker_binaries,
                          nomachine_binaries)
    and not proc.name startswith "runc:"
```

```
    and not java_running_sdjagent
    and not nrpe_becoming_nagios
    and not user_known_non_sudo_setuid_conditions
  output: >
    Unexpected setuid call by non-sudo, non-root program (user=%user.name user_loginuid=%user.loginuid
cur_uid=%user.uid parent=%proc.pname
    command=%proc.cmdline uid=%evt.arg.uid container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [users, mitre_privilege_escalation]
```

30. ユーザ管理バイナリー：User mgmt binaries

ユーザー、パスワード、または許可を管理できるプログラムによる活動。sudo と su は除外されます。コンテナ内の活動も除外されます。コンテナの中には、起動時にベースとなる linux ディストリビューションの上にカスタムユーザを作成するものもあります。実際には何も変更しない無害なコマンドラインは除外されています。

```
- macro: user_known_user_management_activities
  condition: (never_true)

- macro: chage_list
  condition: (proc.name=chage and (proc.cmdline contains "-l" or proc.cmdline contains "--list"))

- rule: User mgmt binaries
  desc: >
    activity by any programs that can manage users, passwords, or permissions. sudo and su are excluded.
    Activity in containers is also excluded--some containers create custom users on top
    of a base linux distribution at startup.
    Some innocuous commandlines that don't actually change anything are excluded.
  condition: >
    spawned_process and proc.name in (user_mgmt_binaries) and
    not proc.name in (su, sudo, lastlog, nologin, unix_chkpwd) and not container and
    not proc.pname in (cron_binaries, systemd, systemd.postins, udev.postinst, run-parts) and
    not proc.cmdline startswith "passwd -S" and
    not proc.cmdline startswith "useradd -D" and
    not proc.cmdline startswith "systemd --version" and
    not run_by_qualys and
    not run_by_sumologic_securefiles and
    not run_by_yum and
    not run_by_ms_oms and
    not run_by_google_accounts_daemon and
    not chage_list and
    not user_known_user_management_activities
  output: >
    User management binary command run outside of container
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline parent=%proc.pname
gparent=%proc.aname[2] ggparent=%proc.aname[3] gggparent=%proc.aname[4])
  priority: NOTICE
  tags: [host, users, mitre_persistence]
```

31. dev以下にファイルを作成する：Create files below dev

デバイスを管理する既知のプログラム以外の /dev 以下にファイルを作成します。ルートキットの中には /dev にファイルを隠すものがあります。

```
- list: allowed_dev_files
  items: [
    /dev/null, /dev/stdin, /dev/stdout, /dev/stderr,
    /dev/random, /dev/urandom, /dev/console, /dev/kmsg
    ]

- macro: user_known_create_files_below_dev_activities
  condition: (never_true)

# (誤検知に対するチェックを追加する必要があるかもしれません。
# https://bugs.launchpad.net/ubuntu/+source/rkhunter/+bug/86153)

- rule: Create files below dev
  desc: creating any files below /dev other than known programs that manage devices. Some rootkits hide files in
/dev.
  condition: >
    fd.directory = /dev and
    (evt.type = creat or ((evt.type = open or evt.type = openat) and evt.arg.flags contains O_CREAT))
    and not proc.name in (dev_creation_binaries)
    and not fd.name in (allowed_dev_files)
    and not fd.name startswith /dev/tty
    and not user_known_create_files_below_dev_activities
  output: "File created below /dev by untrusted program (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline file=%fd.name container_id=%container.id image=%container.image.repository)"
  priority: ERROR
  tags: [filesystem, mitre_persistence]
```

32. コンテナからEC2インスタンスメタデータサービスへの問合せ：Contact EC2 Instance Metadata Service From Container

コンテナからEC2インスタンスメタデータサービスに問合せしようとする試みを検出します。

ローカル/ユーザールールファイルでは、このマクロをオーバーライドして、EC2メタデータへのアクセスを許可したいコンテナイメージを明示的に列挙することができます。このメインの falco ルールファイルでは、アクセスを許可すべきすべてのコンテナを知る方法はないので、"container" マクロを繰り返すことで、どのコンテナも許可されます。オーバーライドされたマクロでは、条件は次のようになります。

(container.image.repository = vendor/container-1 または
container.image.repository = vendor/container-2 or ....)

```
- macro: ec2_metadata_containers
  condition: container
```

EC2インスタンスでは、169.254.169.254はインスタンスに関するメタデータを取得するために使用される特別なIPです。コンテナからこのIPにアクセスできないようにすることが望ましいかもしれません。

```
- rule: Contact EC2 Instance Metadata Service From Container
  desc: Detect attempts to contact the EC2 Instance Metadata Service from a container
  condition: outbound and fd.sip="169.254.169.254" and container and not ec2_metadata_containers
  output: Outbound connection to EC2 instance metadata service (command=%proc.cmdline connection=%fd.name
%container.info image=%container.image.repository:%container.image.tag)
  priority: NOTICE
  tags: [network, aws, container, mitre_discovery]
```

33. コンテナからクラウドメタデータサービスへ問合せ：Contact cloud metadata service from container

コンテナからクラウド・インスタンス・メタデータ・サービスに問合せしようとする試みを検出します。

このルールはクラウド環境(GCP, AWS, Azure)のみを対象としているため、デフォルトでは有効になっていません。このルールを有効にしたい場合は、最初のマクロを上書きしてください。そして、2つ目のマクロを上書きすることで、メタデータへのアクセスを許可したいコンテナをフィルタリングすることができます。

```
- macro: consider_metadata_access
  condition: (never_true)

- macro: user_known_metadata_access
  condition: (k8s.ns.name = "kube-system")
```

GCP、AWS、Azureでは、169.254.169.254は、インスタンスに関するメタデータを取得するために使用される特別なIPです。メタデータは攻撃者による資格情報の取得に使われる可能性があります。

```
- rule: Contact cloud metadata service from container
  desc: Detect attempts to contact the Cloud Instance Metadata Service from a container
  condition: outbound and fd.sip="169.254.169.254" and container and consider_metadata_access and not
user_known_metadata_access
  output: Outbound connection to cloud instance metadata service (command=%proc.cmdline connection=%fd.name
%container.info image=%container.image.repository:%container.image.tag)
  priority: NOTICE
```

```
  tags: [network, container, mitre_discovery]
```

34. コンテナからK8S APIサーバーへの問い合わせ：Contact K8S API Server From Container

コンテナからの K8S API サーバーへの問い合わせを検出します。

ローカル/ユーザールールファイルで、コンテナ内からK8s API Serverとのコンタクトを許可されているネームスペースまたはコンテナイメージをリストアップします。これは、K8s インフラストラクチャー自体がコンテナ内で実行されている場合をカバーする場合があります。

```
- macro: k8s_containers
  condition: >
    (container.image.repository in (gcr.io/google_containers/hyperkube-amd64,
     gcr.io/google_containers/kube2sky, docker.io/sysdig/falco,
     docker.io/sysdig/sysdig, docker.io/falcosecurity/falco,
     sysdig/falco, sysdig/sysdig, falcosecurity/falco) or (k8s.ns.name = "kube-system"))
- macro: k8s_api_server
  condition: (fd.sip.name="kubernetes.default.svc.cluster.local")

- macro: user_known_contact_k8s_api_server_activities
  condition: (never_true)

- rule: Contact K8S API Server From Container
  desc: Detect attempts to contact the K8S API Server from a container
  condition: >
    evt.type=connect and evt.dir=< and
    (fd.typechar=4 or fd.typechar=6) and
    container and
    not k8s_containers and
    k8s_api_server and
    not user_known_contact_k8s_api_server_activities
  output: Unexpected connection to K8s API Server from container (command=%proc.cmdline %container.info
image=%container.image.repository:%container.image.tag connection=%fd.name)
  priority: NOTICE
  tags: [network, k8s, container, mitre_discovery]
```

35. 予期しないK8s NodePort接続：Unexpected K8s NodePort Connection

コンテナから K8 の NodePorts を使用しようとする試みを検出します。

ローカル/ユーザルールファイルで、コンテナ内からNodePortサービスとのコンタクトを許可されているコンテナイメージをリストアップします。これは、K8s インフラストラクチャー自体がコンテナ内で動作している場合をカバーするかもしれません。デフォルトでは、すべてのコンテナがNodePortサービスとのコンタクトを許可されています。

```
- macro: nodeport_containers
  condition: container

- rule: Unexpected K8s NodePort Connection
  desc: Detect attempts to use K8s NodePorts from a container
  condition: (inbound_outbound) and fd.sport >= 30000 and fd.sport <= 32767 and container and not
nodeport_containers
  output: Unexpected K8s NodePort Connection (command=%proc.cmdline connection=%fd.name
container_id=%container.id image=%container.image.repository)
  priority: NOTICE
  tags: [network, k8s, container, mitre_port_knocking]

- list: network_tool_binaries
  items: [nc, ncat, nmap, dig, tcpdump, tshark, ngrep, telnet, mitmproxy, socat, zmap]

- macro: network_tool_procs
  condition: (proc.name in (network_tool_binaries))
```

36. コンテナでパッケージ管理プロセスの起動：Launch Package Management Process in Container

パッケージ管理プロセスをコンテナ内で実行を検出

ローカル/ユーザールールファイルで、コンテナ内のパッケージ管理プロセスの正当な使用にマッチする条件を作成します。例えば、以下のようになります。

- macro: user_known_package_manager_in_container
  condition: proc.cmdline="dpkg -l"

```
- macro: user_known_package_manager_in_container
  condition: (never_true)

# コンテナは不変であることが前提。パッケージ管理はイメージを構築する際に行う必要があります。
- rule: Launch Package Management Process in Container
  desc: Package management process ran inside container
  condition: >
    spawned_process
    and container
    and user.name != "_apt"
    and package_mgmt_procs
    and not package_mgmt_ancestor_procs
    and not user_known_package_manager_in_container
  output: >
    Package management process launched in container (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: ERROR
  tags: [process, mitre_persistence]
```

37. コンテナ内でのNetcatリモートコード実行：Netcat Remote Code Execution in Container

Netcat プログラムは、リモートコードの実行を可能にするコンテナ内で動作します。

```
- rule: Netcat Remote Code Execution in Container
  desc: Netcat Program runs inside container that allows remote code execution
  condition: >
    spawned_process and container and
    ((proc.name = "nc" and (proc.args contains "-e" or proc.args contains "-c")) or
     (proc.name = "ncat" and (proc.args contains "--sh-exec" or proc.args contains "--exec" or proc.args
contains "-e "
                              or proc.args contains "-c " or proc.args contains "--lua-exec"))
    )
  output: >
    Netcat runs inside container that allows remote code execution (user=%user.name user_loginuid=%user.loginuid
    command=%proc.cmdline container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: WARNING
  tags: [network, process, mitre_execution]
```

38. コンテナ内で怪しいネットワークツールの起動：Launch Suspicious Network Tool in Container

コンテナ内で起動したネットワークツールの検出

```
- macro: user_known_network_tool_activities
  condition: (never_true)

- rule: Launch Suspicious Network Tool in Container
  desc: Detect network tools launched inside container
  condition: >
    spawned_process and container and network_tool_procs and not user_known_network_tool_activities
  output: >
    Network tool launched in container (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
parent_process=%proc.pname
    container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: NOTICE
  tags: [network, process, mitre_discovery, mitre_exfiltration]
```

39. ホスト上で不審なネットワークツールを起動：Launch Suspicious Network Tool on Host

このルールはデフォルトでは有効になっていません。有効にしたい場合は、以下のマクロを修正してください。

```
- macro: consider_network_tools_on_host
  condition: (never_true)

- rule: Launch Suspicious Network Tool on Host
```

```
  desc: Detect network tools launched on the host
  condition: >
    spawned_process and
    not container and
    consider_network_tools_on_host and
    network_tool_procs and
    not user_known_network_tool_activities
  output: >
    Network tool launched on host (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
parent_process=%proc.pname)
  priority: NOTICE
  tags: [network, process, mitre_discovery, mitre_exfiltration]

- list: grep_binaries
  items: [grep, egrep, fgrep]

- macro: grep_commands
  condition: (proc.name in (grep_binaries))
```

40. プライベートキーまたはパスワードの検索：Search Private Keys or Passwords

grep でのプライベートキーまたはパスワードの活動を検出します。

```
- list: grep_binaries
  items: [grep, egrep, fgrep]

- macro: grep_commands
  condition: (proc.name in (grep_binaries))

# パスワード/ssh/ユーザーなどの可能性のあるものを検索するための制限の少ない検索。
- macro: grep_more
  condition: (never_true)

- macro: private_key_or_password
  condition: >
    (proc.args icontains "BEGIN PRIVATE" or
     proc.args icontains "BEGIN RSA PRIVATE" or
     proc.args icontains "BEGIN DSA PRIVATE" or
     proc.args icontains "BEGIN EC PRIVATE" or
     (grep_more and
      (proc.args icontains " pass " or
       proc.args icontains " ssh " or
       proc.args icontains " user "))
    )
- rule: Search Private Keys or Passwords
  desc: >
    Detect grep private keys or passwords activity.
  condition: >
    (spawned_process and
     ((grep_commands and private_key_or_password) or
      (proc.name = "find" and (proc.args contains "id_rsa" or proc.args contains "id_dsa")))
    )
```

```
  output: >
    Grep private keys or passwords activities found
    (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline container_id=%container.id
container_name=%container.name
    image=%container.image.repository:%container.image.tag)
  priority:
    WARNING
  tags: [process, mitre_credential_access]
```

41. ログファイルのクリア：Clear Log Activities

重要なログファイルのクリアを検出

```
- list: log_directories
  items: [/var/log, /dev/log]

- list: log_files
  items: [syslog, auth.log, secure, kern.log, cron, user.log, dpkg.log, last.log, yum.log, access_log,
mysql.log, mysqld.log]

- macro: access_log_files
  condition: (fd.directory in (log_directories) or fd.filename in (log_files))

# クリア可能なホワイトリストログファイルのプレースホルダです。
# マクロは（fd.name startswith "/var/log/app1*"）のようにすることを推奨します。
- macro: allowed_clear_log_files
  condition: (never_true)

- macro: trusted_logging_images
  condition: (container.image.repository endswith "splunk/fluentd-hec" or
              container.image.repository endswith "fluent/fluentd-kubernetes-daemonset" or
              container.image.repository endswith "openshift3/ose-logging-fluentd" or
              container.image.repository endswith "containernetworking/azure-npm")
```

42. ログアクティビティのクリア：Clear Log Activities

重要なログファイルのクリアを検出

```
- rule: Clear Log Activities
  desc: Detect clearing of critical log files
  condition: >
    open_write and
    access_log_files and
    evt.arg.flags contains "O_TRUNC" and
    not trusted_logging_images and
    not allowed_clear_log_files
  output: >
    Log files were tampered (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline file=%fd.name
container_id=%container.id image=%container.image.repository)
  priority:
    WARNING
  tags: [file, mitre_defense_evasion]
```

43. ディスクからバルクデータを削除：Remove Bulk Data from Disk

ディスクからバルクデータをクリアするために実行中のプロセスを検出

```
- list: data_remove_commands
  items: [shred, mkfs, mke2fs]

- macro: clear_data_procs
  condition: (proc.name in (data_remove_commands))

- macro: user_known_remove_data_activities
  condition: (never_true)

- rule: Remove Bulk Data from Disk
  desc: Detect process running to clear bulk data from disk
  condition: spawned_process and clear_data_procs and not user_known_remove_data_activities
  output: >
    Bulk data has been removed from disk (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
file=%fd.name container_id=%container.id image=%container.image.repository)
  priority:
    WARNING
  tags: [process, mitre_persistence]
```

44. シェルヒストリーを削除または名前を変更：Delete or rename shell history

シェルヒストリー削除を検出

```
- macro: modify_shell_history
  condition: >
    (modify and (
      evt.arg.name contains "bash_history" or
      evt.arg.name contains "zsh_history" or
      evt.arg.name contains "fish_read_history" or
      evt.arg.name endswith "fish_history" or
      evt.arg.oldpath contains "bash_history" or
      evt.arg.oldpath contains "zsh_history" or
      evt.arg.oldpath contains "fish_read_history" or
      evt.arg.oldpath endswith "fish_history" or
      evt.arg.path contains "bash_history" or
      evt.arg.path contains "zsh_history" or
      evt.arg.path contains "fish_read_history" or
      evt.arg.path endswith "fish_history"))
- macro: truncate_shell_history
  condition: >
    (open_write and (
      fd.name contains "bash_history" or
      fd.name contains "zsh_history" or
      fd.name contains "fish_read_history" or
      fd.name endswith "fish_history") and evt.arg.flags contains "O_TRUNC")
- macro: var_lib_docker_filepath
```

```
  condition: (evt.arg.name startswith /var/lib/docker or fd.name startswith /var/lib/docker)

- rule: Delete or rename shell history
  desc: Detect shell history deletion
  condition: >
    (modify_shell_history or truncate_shell_history) and
       not var_lib_docker_filepath and
       not proc.name in (docker_binaries)
  output: >
    Shell history had been deleted or renamed (user=%user.name user_loginuid=%user.loginuid type=%evt.type
command=%proc.cmdline fd.name=%fd.name name=%evt.arg.name path=%evt.arg.path oldpath=%evt.arg.oldpath
%container.info)
  priority:
    WARNING
  tags: [process, mitre_defense_evasion]
```

45. Bashヒストリーの削除：Delete Bash History

bashヒストリー削除を検出

このルールは非推奨であり、決してトリガされるべきではありません。バックポートとの互換性を保つために、ここに置いておいてください。ルール シェルの履歴を削除したり名前を変更したりするのは、現在ではこのルールを使うのが好ましいでしょう。

```
- rule: Delete Bash History
  desc: Detect bash history deletion
  condition: >
    ((spawned_process and proc.name in (shred, rm, mv) and proc.args contains "bash_history") or
     (open_write and fd.name contains "bash_history" and evt.arg.flags contains "O_TRUNC"))
  output: >
    Shell history had been deleted or renamed (user=%user.name user_loginuid=%user.loginuid type=%evt.type
command=%proc.cmdline fd.name=%fd.name name=%evt.arg.name path=%evt.arg.path oldpath=%evt.arg.oldpath
%container.info)
  priority:
    WARNING
  tags: [process, mitre_defense_evasion]

- macro: consider_all_chmods
  condition: (always_true)

- list: user_known_chmod_applications
  items: [hyperkube, kubelet]
```

46. Setuid またはSetgid ビットを設定：Set Setuid or Setgid bit

このマクロは、必要に応じてユーザールールでオーバーライドする必要があります。これは、与えられたアプリケーションが user_known_chmod_applications リストで完全に無視されるべきではなく、特定の条件でのみ無視されるべきである場合に便利です。

```
- macro: user_known_set_setuid_or_setgid_bit_conditions
  condition: (never_true)

- rule: Set Setuid or Setgid bit
  desc: >
    When the setuid or setgid bits are set for an application,
    this means that the application will run with the privileges of the owning user or group respectively.
    Detect setuid or setgid bits set via chmod
  condition: >
    consider_all_chmods and chmod and (evt.arg.mode contains "S_ISUID" or evt.arg.mode contains "S_ISGID")
    and not proc.name in (user_known_chmod_applications)
    and not exe_running_docker_save
    and not user_known_set_setuid_or_setgid_bit_conditions
  output: >
    Setuid or setgid bit is set via chmod (fd=%evt.arg.fd filename=%evt.arg.filename mode=%evt.arg.mode
user=%user.name user_loginuid=%user.loginuid process=%proc.name
    command=%proc.cmdline container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority:
    NOTICE
  tags: [process, mitre_persistence]
```

47. 隠しファイルやディレクトリの作成：Create Hidden Files or Directories

作成された隠しファイルやディレクトリを検出

```
- list: exclude_hidden_directories
  items: [/root/.cassandra]

# このルールを使用するには、consider_hidden_file_creationを修正する必要があります。
- macro: consider_hidden_file_creation
  condition: (never_true)

- macro: user_known_create_hidden_file_activities
  condition: (never_true)

- rule: Create Hidden Files or Directories
  desc: Detect hidden files or directories created
  condition: >
    ((modify and evt.arg.newpath contains "/.") or
     (mkdir and evt.arg.path contains "/.") or
     (open_write and evt.arg.flags contains "O_CREAT" and fd.name contains "/." and not fd.name pmatch
(exclude_hidden_directories))) and
    consider_hidden_file_creation and
    not user_known_create_hidden_file_activities
  output: >
    Hidden file or directory created (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
    file=%fd.name newpath=%evt.arg.newpath container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority:
    NOTICE
  tags: [file, mitre_persistence]
```

48. コンテナでリモートファイルコピーツールを起動：Launch Remote File Copy Tools in Container

コンテナ内で起動したリモートファイルコピーツールを検出

```
- list: remote_file_copy_binaries
  items: [rsync, scp, sftp, dcp]

- macro: remote_file_copy_procs
  condition: (proc.name in (remote_file_copy_binaries))

# ユーザーはこのマクロを上書きして、コンテナ内でリモートファイルコピーツールを
# 使用するためのカスタム条件を指定する必要があります。
- macro: user_known_remote_file_copy_activities
  condition: (never_true)

- rule: Launch Remote File Copy Tools in Container
  desc: Detect remote file copy tools launched in container
  condition: >
    spawned_process
    and container
    and remote_file_copy_procs
    and not user_known_remote_file_copy_activities
  output: >
    Remote file copy tool launched in container (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline parent_process=%proc.pname
    container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: NOTICE
  tags: [network, process, mitre_lateral_movement, mitre_exfiltration]

- rule: Create Symlink Over Sensitive Files
  desc: Detect symlink created over sensitive files
  condition: >
    create_symlink and
    (evt.arg.target in (sensitive_file_names) or evt.arg.target in (sensitive_directory_names))
  output: >
    Symlinks created over senstivie files (user=%user.name user_loginuid=%user.loginuid command=%proc.cmdline
target=%evt.arg.target linkpath=%evt.arg.linkpath parent_process=%proc.pname)
  priority: NOTICE
  tags: [file, mitre_exfiltration]
```

49. 一般的なマイナープールポートへのアウトバウンド接続を検出：Detect outbound connections to common miner pool ports

マイナーは通常、一般的なポートでマイナープールに接続します。

```
- list: miner_ports
  items: [
        25, 3333, 3334, 3335, 3336, 3357, 4444,
        5555, 5556, 5588, 5730, 6099, 6666, 7777,
        7778, 8000, 8001, 8008, 8080, 8118, 8333,
        8888, 8899, 9332, 9999, 14433, 14444,
```

```
        45560, 45700
    ]

- list: miner_domains
  items: [
      "asia1.ethpool.org","ca.minexmr.com",
      "cn.stratum.slushpool.com","de.minexmr.com",
      "eth-ar.dwarfpool.com","eth-asia.dwarfpool.com",
      "eth-asia1.nanopool.org","eth-au.dwarfpool.com",
      "eth-au1.nanopool.org","eth-br.dwarfpool.com",
      "eth-cn.dwarfpool.com","eth-cn2.dwarfpool.com",
      "eth-eu.dwarfpool.com","eth-eu1.nanopool.org",
      "eth-eu2.nanopool.org","eth-hk.dwarfpool.com",
      "eth-jp1.nanopool.org","eth-ru.dwarfpool.com",
      "eth-ru2.dwarfpool.com","eth-sg.dwarfpool.com",
      "eth-us-east1.nanopool.org","eth-us-west1.nanopool.org",
      "eth-us.dwarfpool.com","eth-us2.dwarfpool.com",
      "eu.stratum.slushpool.com","eu1.ethermine.org",
      "eu1.ethpool.org","fr.minexmr.com",
      "mine.moneropool.com","mine.xmrpool.net",
      "pool.minexmr.com","pool.monero.hashvault.pro",
      "pool.supportxmr.com","sg.minexmr.com",
      "sg.stratum.slushpool.com","stratum-eth.antpool.com",
      "stratum-ltc.antpool.com","stratum-zec.antpool.com",
      "stratum.antpool.com","us-east.stratum.slushpool.com",
      "us1.ethermine.org","us1.ethpool.org",
      "us2.ethermine.org","us2.ethpool.org",
      "xmr-asia1.nanopool.org","xmr-au1.nanopool.org",
      "xmr-eu1.nanopool.org","xmr-eu2.nanopool.org",
      "xmr-jp1.nanopool.org","xmr-us-east1.nanopool.org",
      "xmr-us-west1.nanopool.org","xmr.crypto-pool.fr",
      "xmr.pool.minergate.com"
      ]

- list: https_miner_domains
  items: [
    "ca.minexmr.com",
    "cn.stratum.slushpool.com",
    "de.minexmr.com",
    "fr.minexmr.com",
    "mine.moneropool.com",
    "mine.xmrpool.net",
    "pool.minexmr.com",
    "sg.minexmr.com",
    "stratum-eth.antpool.com",
    "stratum-ltc.antpool.com",
    "stratum-zec.antpool.com",
    "stratum.antpool.com",
    "xmr.crypto-pool.fr"
  ]

- list: http_miner_domains
  items: [
    "ca.minexmr.com",
    "de.minexmr.com",
```

```
    "fr.minexmr.com",
    "mine.moneropool.com",
    "mine.xmrpool.net",
    "pool.minexmr.com",
    "sg.minexmr.com",
    "xmr.crypto-pool.fr"
  ]

# クリプトマイニングのIOCに基づいてルールを追加する

- macro: minerpool_https
  condition: (fd.sport="443" and fd.sip.name in (https_miner_domains))

- macro: minerpool_http
  condition: (fd.sport="80" and fd.sip.name in (http_miner_domains))

- macro: minerpool_other
  condition: (fd.sport in (miner_ports) and fd.sip.name in (miner_domains))

- macro: net_miner_pool
  condition: (evt.type in (sendto, sendmsg) and evt.dir=< and (fd.net != "127.0.0.0/8" and not fd.snet in
(rfc_1918_addresses)) and ((minerpool_http) or (minerpool_https) or (minerpool_other)))

- macro: trusted_images_query_miner_domain_dns
  condition: (container.image.repository in (docker.io/falcosecurity/falco, falcosecurity/falco))
  append: false

# このルールはデフォルトでは無効になっています。
# 注意: falcoは、あなたの環境でアラートを引き起こす可能性のあるマイナープールドメインを解決するためにDNSリクエストを送信します。
- rule: Detect outbound connections to common miner pool ports
  desc: Miners typically connect to miner pools on common ports.
  condition: net_miner_pool and not trusted_images_query_miner_domain_dns
  enabled: false
  output: Outbound connection to IP/Port flagged by cryptoioc.ch (command=%proc.cmdline port=%fd.rport
ip=%fd.rip container=%container.info image=%container.image.repository)
  priority: CRITICAL
  tags: [network, mitre_execution]
```

50. Stratumプロトコルを使用したクリプトマイナーの検出：Detect crypto miners using the Stratum protocol

マイナーは通常、'stratum+tcp'で始まるURIで接続先のマイニングプールを指定します。

```
- rule: Detect crypto miners using the Stratum protocol
  desc: Miners typically specify the mining pool to connect to with a URI that begins with 'stratum+tcp'
  condition: spawned_process and proc.cmdline contains "stratum+tcp"
  output: Possible miner running (command=%proc.cmdline container=%container.info
image=%container.image.repository)
  priority: CRITICAL
  tags: [process, mitre_execution]
```

51. コンテナ内でdockerクライアントが実行された：The docker client is executed in a container

コンテナ内でk8s クライアントツールの実行を検出

```
- list: k8s_client_binaries
  items: [docker, kubectl, crictl]

- list: user_known_k8s_ns_kube_system_images
  items: [
      k8s.gcr.io/fluentd-gcp-scaler,
      k8s.gcr.io/node-problem-detector/node-problem-detector
  ]

- list: user_known_k8s_images
  items: [
      mcr.microsoft.com/aks/hcp/hcp-tunnel-front
  ]

# Whitelist for known docker client binaries run inside container
# - k8s.gcr.io/fluentd-gcp-scaler in GCP/GKE
- macro: user_known_k8s_client_container
  condition: >
    (k8s.ns.name="kube-system" and container.image.repository in (user_known_k8s_ns_kube_system_images)) or
container.image.repository in (user_known_k8s_images)
- macro: user_known_k8s_client_container_parens
  condition: (user_known_k8s_client_container)

- rule: The docker client is executed in a container
  desc: Detect a k8s client tool executed inside a container
  condition: spawned_process and container and not user_known_k8s_client_container_parens and proc.name in
(k8s_client_binaries)
  output: "Docker or kubernetes client executed in container (user=%user.name user_loginuid=%user.loginuid
%container.info parent=%proc.pname cmdline=%proc.cmdline
image=%container.image.repository:%container.image.tag)"
  priority: WARNING
  tags: [container, mitre_execution]
```

52. コンテナ内でパケットソケットが作成された：Packet socket created in container

コンテナ内のデバイスドライバ(OSIレイヤ2)レベルでパケットソケットを検出します。パケットソケットは、攻撃者による ARPスプーフィングや特権昇格 (CVE-2020-14386) に利用される可能性があります。

```
# このルールはデフォルトで有効になっています。
# 無効にしたい場合は、以下のマクロを修正します。
- macro: consider_packet_socket_communication
  condition: (always_true)

- list: user_known_packet_socket_binaries
  items: []

- rule: Packet socket created in container
```

```
  desc: Detect new packet socket at the device driver (OSI Layer 2) level in a container. Packet socket could be
used for ARP Spoofing and privilege escalation(CVE-2020-14386) by attacker.
  condition: evt.type=socket and evt.arg[0]=AF_PACKET and consider_packet_socket_communication and container and
not proc.name in (user_known_packet_socket_binaries)
  output: Packet socket was created in a container (user=%user.name user_loginuid=%user.loginuid
command=%proc.cmdline socket_info=%evt.args container_id=%container.id container_name=%container.name
image=%container.image.repository:%container.image.tag)
  priority: NOTICE
  tags: [network, mitre_discovery]
```

53. ローカルサブネット外のネットワーク接続：Network Connection outside Local Subnet

ローカルサブネット外のイメージへのトラフィックを検出します。

```
# ローカルサブネット外のネットワーク接続」ルールを有効にするには、(always_true)に変更します。
- macro: enabled_rule_network_only_subnet
  condition: (never_true)

# アウトバウンドトラフィックを持つことを許可されているイメージ
- list: images_allow_network_outside_subnet
  items: []

# ルールが適用されるネームスペース
- list: namespace_scope_network_only_subnet
  items: []

- macro: network_local_subnet
  condition: >
    fd.rnet in (rfc_1918_addresses) or
    fd.ip = "0.0.0.0" or
    fd.net = "127.0.0.0/8"
# # テストの方法:
# # マクロenabled_rule_network_only_subnetを条件: always_trueに変更します
# # namespace_scope_network_only_subnetに'default'を追加する
# # Run:
# kubectl run --generator=run-pod/v1 -n default -i --tty busybox --image=busybox --rm -- wget google.com -O
/var/google.html
# # 実行中のログをチェック

- rule: Network Connection outside Local Subnet
  desc: Detect traffic to image outside local subnet.
  condition: >
    enabled_rule_network_only_subnet and
    inbound_outbound and
    container and
    not network_local_subnet and
    k8s.ns.name in (namespace_scope_network_only_subnet)
  output: >
    Network connection outside local subnet
    (command=%proc.cmdline connection=%fd.name user=%user.name user_loginuid=%user.loginuid
container_id=%container.id
     image=%container.image.repository namespace=%k8s.ns.name
     fd.rip.name=%fd.rip.name fd.lip.name=%fd.lip.name fd.cip.name=%fd.cip.name fd.sip.name=%fd.sip.name)
  priority: WARNING
```

```
  tags: [network]
```

54. アウトバウンドまたはインバウンドトラフィックが許可されたサーバーのプロセスとポートで無い：Outbound or Inbound Traffic not to Authorized Server Process and Port

許可されたサーバプロセスとポート以外のトラフィックを検出します。

```
- macro: allowed_port
  condition: (never_true)

- list: allowed_image
  items: [] # 監視にイメージを追加, 例: bitnami/nginx

- list: authorized_server_binaries
  items: []  # 許可するバイナリーへ追加, 例: nginx

- list: authorized_server_port
  items: [] # 許可するポートを追加, 例: 80

# # テストの方法:
# kubectl run --image=nginx nginx-app --port=80 --env="DOMAIN=cluster"
# kubectl expose deployment nginx-app --port=80 --name=nginx-http --type=LoadBalancer
# # On minikube:
# minikube service nginx-http
# # On general K8s:
# kubectl get services
# kubectl cluster-info
# # Nginxのサービスとポートを参照してください。
# # ルールを別のポートに変更し、プロセス名を変更して、再度テストしてみてください。

- rule: Outbound or Inbound Traffic not to Authorized Server Process and Port
  desc: Detect traffic that is not to authorized server process and port.
  condition: >
    allowed_port and
    inbound_outbound and
    container and
    container.image.repository in (allowed_image) and
    not proc.name in (authorized_server_binary) and
    not fd.sport in (authorized_server_port)
  output: >
    Network connection outside authorized port and binary
    (command=%proc.cmdline connection=%fd.name user=%user.name user_loginuid=%user.loginuid
container_id=%container.id
    image=%container.image.repository)
  priority: WARNING
  tags: [network]
```

55. コンテナ内のSTDOUT/STDINをネットワーク接続にリダイレクトする：Redirect STDOUT/STDIN to Network Connection in Container

コンテナ内のネットワーク接続への stdout/stdin のリダイレクトを検出 (リバースシェルの可能性あり)

```
- macro: user_known_stand_streams_redirect_activities
  condition: (never_true)

- rule: Redirect STDOUT/STDIN to Network Connection in Container
  desc: Detect redirecting stdout/stdin to network connection in container (potential reverse shell).
  condition: evt.type=dup and evt.dir=> and container and fd.num in (0, 1, 2) and fd.type in ("ipv4", "ipv6")
and not user_known_stand_streams_redirect_activities
  output: >
    Redirect stdout/stdin to network connection (user=%user.name user_loginuid=%user.loginuid %container.info
process=%proc.name parent=%proc.pname cmdline=%proc.cmdline terminal=%proc.tty container_id=%container.id
image=%container.image.repository fd.name=%fd.name fd.num=%fd.num fd.type=%fd.type fd.sip=%fd.sip)
  priority: WARNING
```

56. コンテナドリフト検出 (chmod)：Container Drift Detected (chmod)

chmodによりコンテナ内に作成された新しい実行ファイル

コンテナ内に新しい実行ファイルが作成されると、以下の2つのコンテナドリフトルールが発動します。実行ファイルを作成するには、実行権限を持ったファイルを作成する方法と、既存のファイルの権限を変更して作成する方法の2つの方法があります。新しい sysdig フィルタ is_open_exec を使用して、実行権限のあるファイルが作成されたかどうかを調べ、コンテナ内の全ての chmod をトレースします。ここでターゲットとしているユースケースは、コンテナの一部としてシップされていないコードを実行しようとする試みです（ドリフト） - 悪意のある、あるいはコンプライアンスに準拠していない可能性のある活動です。

2つの注意点があります。

1) ほとんどの場合、'docker cp' は特定されませんが、攻撃者がコンテナのランタイムデーモンにアクセスした場合、すでに特権を持っていると仮定しています。

2) コンテナがビルドされている環境ではドリフトルールがノイズになる（dockerビルドなど

これら2つのルールはデフォルトでは有効になっていません。マクロ条件で `never_true` を使用して有効にします。

```
- macro: user_known_container_drift_activities
```

```
  condition: (always_true)

- rule: Container Drift Detected (chmod)
  desc: New executable created in a container due to chmod
  condition: >
    chmod and
    consider_all_chmods and
    container and
    not runc_writing_exec_fifo and
    not runc_writing_var_lib_docker and
    not user_known_container_drift_activities and
    evt.rawres>=0 and
    ((evt.arg.mode contains "S_IXUSR") or
    (evt.arg.mode contains "S_IXGRP") or
    (evt.arg.mode contains "S_IXOTH"))
  output: Drift detected (chmod), new executable created in a container (user=%user.name
user_loginuid=%user.loginuid command=%proc.cmdline filename=%evt.arg.filename name=%evt.arg.name
mode=%evt.arg.mode event=%evt.type)
  priority: ERROR
```

57. コンテナドリフト検出(open+create)：Container Drift Detected (open+create)

open+createによりコンテナ内に作成された新しい実行ファイル

Container Drift Detected (open+create)にはFALCO_ENGINE_VERSION 6が必要です。

```
- rule: Container Drift Detected (open+create)
  desc: New executable created in a container due to open+create
  condition: >
    evt.type in (open,openat,creat) and
    evt.is_open_exec=true and
    container and
    not runc_writing_exec_fifo and
    not runc_writing_var_lib_docker and
    not user_known_container_drift_activities and
    evt.rawres>=0
  output: Drift detected (open+create), new executable created in a container (user=%user.name
user_loginuid=%user.loginuid command=%proc.cmdline filename=%evt.arg.filename name=%evt.arg.name
mode=%evt.arg.mode event=%evt.type)
  priority: ERROR
```

58. C2サーバへのアウトバウンド接続：Outbound Connection to C2 Servers

コマンド＆コントロールサーバへのアウトバウンド接続を検出

```
- list: c2_server_ip_list
  items: []

- rule: Outbound Connection to C2 Servers
  desc: Detect outbound connection to command & control servers
  condition: outbound and fd.sip in (c2_server_ip_list)
```

```
  output: Outbound connection to C2 server (command=%proc.cmdline connection=%fd.name
user=%user.name user_loginuid=%user.loginuid container_id=%container.id
image=%container.image.repository)
  priority: WARNING
 tags: [network]
```

59. Linuxカーネルモジュールのインジェクションを検出：Linux Kernel Module Injection Detected

(コンテナから)カーネルモジュールがインジェクトされたことを検出しました。

```
- list: white_listed_modules
  items: []

- rule: Linux Kernel Module Injection Detected
  desc: Detect kernel module was injected (from container).
  condition: spawned_process and container and proc.name=insmod and not proc.args in (white_listed_modules)
  output: Linux Kernel Module injection using insmod detected (user=%user.name user_loginuid=%user.loginuid
parent_process=%proc.pname module=%proc.args)
  priority: WARNING
  tags: [process]
```

60. コンテナをルートユーザとして実行：Container Run as Root User

root ユーザーとして動作しているコンテナを検出

このルールはデフォルトでは無効になっており、非ルートコンテナポリシーが適用されている場合は有効にする必要があります。ユーザネームスペースが適用されている場合、例えば userns-remap が有効になっている場合などは、このルールは期待通りには動作しないことに注意してください。

```
- list: run_as_root_image_list
  items: []

- macro: user_known_run_as_root_container
  condition: (container.image.repository in (run_as_root_image_list))

# このルールはデフォルトでは無効になっており、非ルートコンテナポリシーが適用されている場合は有効にする必要があります。
#  ユーザネームスペースが適用されている場合、例えば userns-remap が有効になっている場合などは、このルールは期待通りには動
作しないことに注意してください。
- rule: Container Run as Root User
  desc: Detected container running as root user
  condition: spawned_process and container and proc.vpid=1 and user.uid=0 and not
user_known_run_as_root_container
  enabled: false
  output: Container launched with root user privilege (uid=%user.uid container_id=%container.id
container_name=%container.name image=%container.image.repository:%container.image.tag)
  priority: INFO
  tags: [container, process]
```

アプリケーションルールはapplication_rules.yamlに移動しました。falco_rules.local.yamlに追加して有効にしたい場合はそちらをご覧ください。