WTF! My Container Just Spawned A Shell!

Mark Stemm,
Sysdig/Falco Engineer
Home Security Analogy

- Prevent Intrusion
  - Door locks
  - Window sensors
  - Bars on ground floor windows
  - Exterior camera
Home Security Analogy

- Detect Intrusion
  - Motion sensors
  - Interior cameras
Home Security Analogy

- Prevent Intrusion
  - Passwords
  - Two-Factor Authentication
  - Fixing Software Vulnerabilities
  - Container Image Scanning
  - Firewalls
Home Security Analogy

- Detect Intrusion
  - Sysdig Falco!
- Both methods essential for full protection
What is Sysdig Falco

• A behavioral activity monitor
• Detects suspicious activity defined by a set of rules
• Uses sysdig’s flexible and powerful filtering expressions
• With full support for containers/orchestration
• Utilizes sysdig’s container support
• And flexible notification methods
• Alert to files, standard output, syslog, programs
• Open Source
• Anyone can contribute rules or improvements
Quick Examples

- **A shell is run in a container**
  - `container.id != host` and `proc.name = bash`

- **Overwrite system binaries**
  - `fd.directory` in `(/bin, /sbin, /usr/bin, /usr/sbin)` and write

- **Container namespace change**
  - `evt.type = setns` and not `proc.name` in `(docker, sysdig)`

- **Non-device files written in /dev**
  - `(evt.type = creat or evt.arg.flags contains O_CREAT)` and `proc.name != blkid` and `fd.directory = /dev` and `fd.name != /dev/null`

- **Process tries to access camera**
  - `evt.type = open` and `fd.name = /dev/video0` and not `proc.name` in `(skype, webex)`
Falco Rules

- `.yml` file containing Macros, Lists, and Rules
  - Example:

```yaml
- macro: bin_dir
  condition: fd.directory in (/bin, /sbin, /usr/bin, /usr/sbin)

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

- rule: write_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
  output: "File below a known binary directory opened for writing (user=%user.name command=%proc.cmdline file=%fd.name)"
  priority: WARNING
```
Falco Rules

- Macros
  - **name**: text to use in later rules
  - **condition**: filter expression snippet
- List
  - **name**: text to use later
  - **items**: list of items
- Rules
  - **name**: used to identify rule
  - **desc**: description of rule
  - **condition**: filter expression, can contain macro references
  - **output**: message to emit when rule triggers, can contain formatted info from event
  - **priority**: severity of rule (WARNING, INFO, etc.)
Falco Rules

- Filtering Expressions
  - Use the same format as sysdig
  - Full container/k8s/mesos/etc support
  - Falco rules are combined into one giant filtering expression, joined by ORs
  - Each rule must contain at least one `evt.type` expression
    - i.e. `evt.type=open` and ...
    - Allows for very fast filtering of events.
Alerts And Outputs

- Events that match filter expression result in alerts
- Rule’s `output` field used to format event into alert message
- Falco configuration used to control where alert message is sent.
- Any combination of
  - Syslog
  - File
  - Standard Output
  - Shell (e.g. `mail -s "Falco Notification" someone@example.com`)
Installing Falco

• Debian Package
  • `apt-get -y install falco`
• Redhat Package
  • `yum -y install falco`
• Installation Script
  • `curl -s https://s3.amazonaws.com/download.draios.com/stable/install-falco`
• Docker container
  • `docker pull sysdig/falco`
Running Falco

- As a service
  - `$ service falco start`
  - alerts to syslog
- By hand
  - `$ sudo falco -r <rules file> -c <config file>`
  - alerts to syslog, stdout
- Using docker
  - `docker run -i -t --name falco --privileged -v /var/run/docker.sock:/host/var/run/docker.sock -v /dev:/host/dev -v /proc:/host/proc:ro -v /boot:/host/boot:ro -v /lib/modules:/host/lib/modules:ro -v /usr:/host/usr:ro sysdig/falco`
Demo
What we’re going to show you

- Falco installation using docker
- Walkthrough of simple attacks
  - Writing to files below /bin
- Synthetic event generator
- Badly configured containers
- Exploiting a bad REST API
- Exploiting PHPwebmail vulnerability
Join The Community

- Website
  - http://www.sysdig.org/falco/
- Mailing List
  - https://groups.google.com/forum/#!forum/falco
- Public Slack
  - https://sysdig.slack.com/messages/falco/
- Blog
  - https://sysdig.com/blog/tag/falco/
- Sysdig Monitor
  - http://sysdig.com/product
Learn More

- Github
  - https://github.com/draios/falco
  - Pull Requests welcome!
- Wiki
  - https://github.com/draios/falco/wiki
- Docker Hub
  - https://hub.docker.com/r/sysdig/falco/
Thank You!