AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

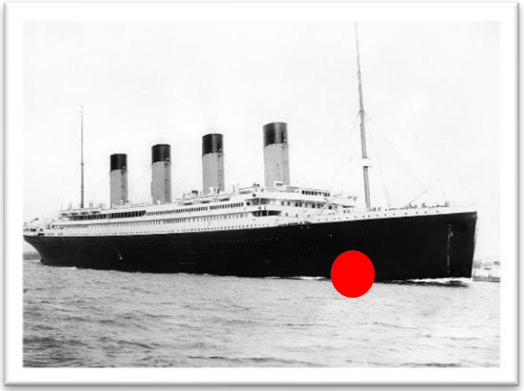
DON'T GET HACKED, GET AMINER

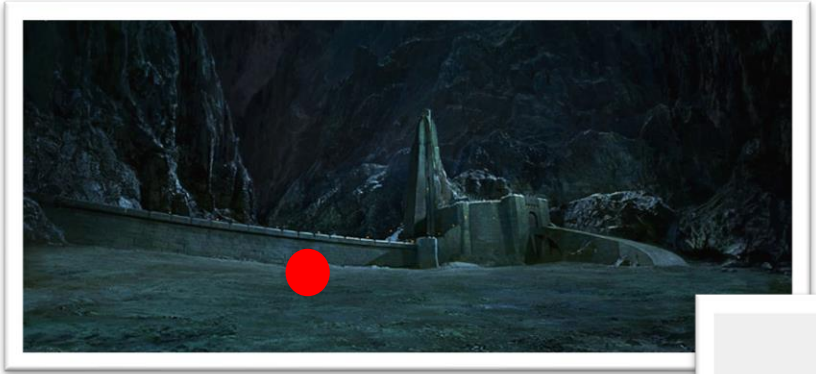
Log Data Analysis for Intrusion Detection

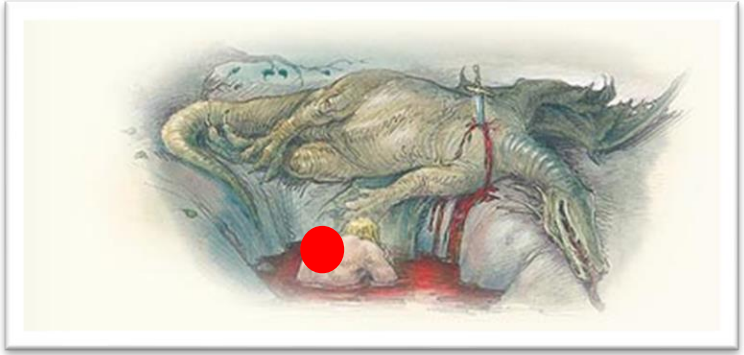
Florian Skopik, Markus Wurzenberger, Max Landauer

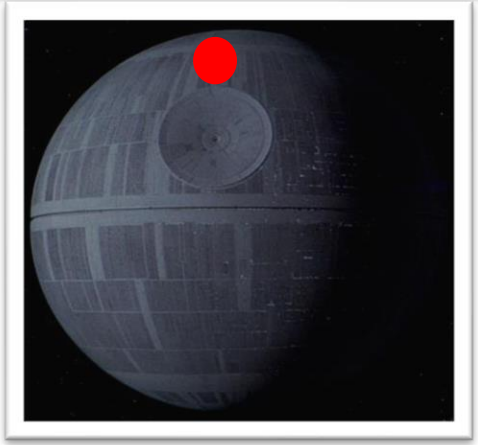
In-Depth Security Conference Europe (DeepSec), 19.11.2021

WHAT DO THEY HAVE IN COMMON?



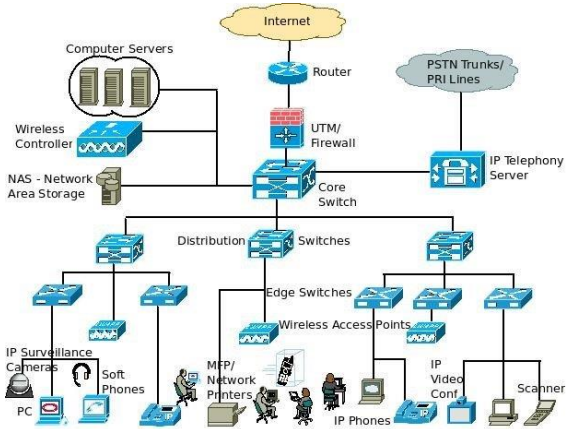






MONITORING TODAY

- Complex systems organically grow bottom up
  - Implementation and configuration failures lead to vulnerabilities
  - Design errors cause weaknesses
  - **Prevention ultimately fails**
- Monitoring focuses on the early **discovery of adversarial actions**, such as the exploitation of vulnerabilities and weaknesses
- Today‘s **state of the art** still is:
  - Mainly **investigation of network traffic** (common tools)
  - **Signature-based** search for known bad
  - Log data investigation with **SIEMs** – mostly predefined rules only
  - Limited anomaly detection (mostly outliers, e.g. uncommon protocols)

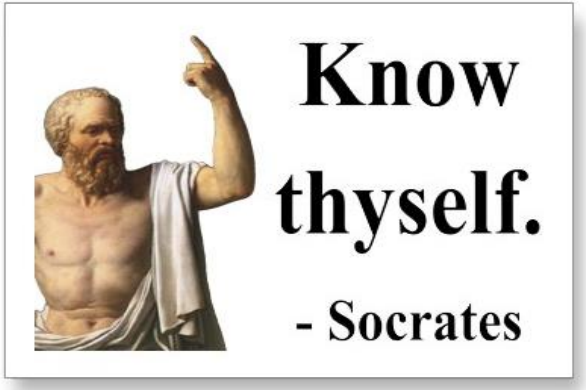


ISSUES WITH NETWORK SECURITY MONITORING

- Increasing use of **encryption** drastically **limits** the **visibility** on the network
  - TLS everywhere
  - Even for DNS: DoH, DoT on the horizon
  - Focus on limited initial handshake (TLS 1.3), netflows
- Attackers „**living off the land**“
  - Use of built-in tools (cmd.exe, powershell.exe etc.)
  - Use of standard protocols
- High-profile attacks often do not exploit any technical vulnerabilities at all
  - **Social engineering**
  - Compromise of legitimate update mechanisms, root certificates etc.
  - Hence, **no signature** can capture that behavior, **no rule** can flag malicious activities

A WAY OUT: ANOMALY DETECTION

- However, **attackers utilize systems differently** from legitimate users…
  - Access other DMZ servers from a compromised web server
  - Use of SSH maintenance interface instead of the web interface
  - Login using backup system SSH key intended for SFTP file transfer
- Novel **machine learning** approaches
  - **Observe a system** and its “normal” utilization
  - Dynamically **build up** a model that constitutes a **baseline**
  - **Alert on** significant **deviations** from this baseline
- **Visibility** of adversarial actions is key!
  - On the **endpoint**
  - Use what we have – no additional agent
  - **Verbose log data** from services, application, operating systems



INTRUSION DETECTION: TECHNIQUES OVERVIEW

Signature-based detection

Blocked:

- 192.168.141.10
- 192.168.176.23

Monitored logs:

- 10.237.2.50
- 10.237.2.22
- 10.237.2.50
- 192.168.175.131
- 192.168.176.23 

+ Efficient
- Only known bad values
- Variants

Allowlisting

Allowed:

- 10.237.2.50
- 172.28.193.48
- 10.237.2.22

Monitored logs:

- 10.237.2.50
- 10.237.2.22
- 10.237.2.50
- 192.168.175.131 
- 192.168.176.23 

+ Exact
+ Detects unknown attacks
- Complex to generate/maintain

Anomaly Detection

Monitored logs:

- 10.237.2.50 
- 10.237.2.22 
- 10.237.2.50
- 172.28.195.6 
- 172.28.193.48 
- 10.237.2.22

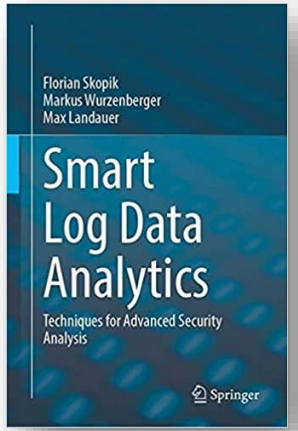
- 10.237.2.50
- 172.28.194.6 
- 172.28.193.48

+ Detects unknown attacks
+ Self-learning
- Training phase (benign behavior)
- False positives

MAKE LOG DATA ANALYSIS SMART!

- Log data are **textual data,** not simple numerical values
- Log data have **mostly unknown structure** and unknown meaning
- For intrusion detection, log data need to be **processed online** („single pass approaches“)
- Observed systems change frequently (updates, extensions, etc.), leading to a **moving baseline**.

- **SOLUTIONS in this presentation**: Machine Learning and AI for …
  - Part I: Flexible creation of log data parsers
  - Part II: Online anomaly detection beyond simple outlier detection



https://www.amazon.de/dp/3030744493

COMMON LOG FORMATS

- **Apache Access logs**
  - Structured
  - Many Categorical variables
  - Some complex tokens (session Ids, …)

```
10.35.34.242 - - [04/Oct/2021:07:10:46 +0000] "GET / HTTP/1.1" 200 6122 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:86.0) Gecko/20100101 Firefox/86.0"
10.35.35.75 - - [04/Oct/2021:07:10:48 +0000] "POST /wp-cron.php?doing_wp_cron=1633331448.0809569358825683593750 HTTP/1.1" 200 150 "-" "WordPress/5.8.1; https://intranet.price.fox.org"
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-includes/css/dist/block-library/style.min.css?ver=5.8.1 HTTP/1.1" 200 10846 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; L
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-content/themes/go/dist/css/style-shared.min.css?ver=1.4.4 HTTP/1.1" 200 23724 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu;
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-content/themes/go/dist/css/design-styles/style-traditional.min.css?ver=1.4.4 HTTP/1.1" 200 1490 "http://intranet.price.fox.org/" "Mozilla
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-includes/js/wp-embed.min.js?ver=5.8.1 HTTP/1.1" 200 1099 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:86.
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2 HTTP/1.1" 200 4505 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux x
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-includes/js/wp-emoji-release.min.js?ver=5.8.1 HTTP/1.1" 200 5266 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-content/themes/go/dist/js/frontend.min.js?ver=1.4.4 HTTP/1.1" 200 11448 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux
10.35.34.242 - - [04/Oct/2021:07:10:48 +0000] "GET /wp-includes/js/jquery/jquery.min.js?ver=3.6.0 HTTP/1.1" 200 31245 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
10.35.34.242 - - [04/Oct/2021:07:10:49 +0000] "GET /favicon.ico HTTP/1.1" 404 396 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:86.0) Gecko/20100101 Firefox/86.0
::1 - - [04/Oct/2021:07:10:54 +0000] "OPTIONS * HTTP/1.0" 200 110 "-" "Apache/2.4.29 (Ubuntu) OpenSSL/1.1.1 (internal dummy connection)"
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET / HTTP/1.1" 200 6122 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/93.0.4577.63 Safari/537.36"
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-includes/css/dist/block-library/style.min.css?ver=5.8.1 HTTP/1.1" 200 10846 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x86
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-content/themes/go/dist/css/design-styles/style-traditional.min.css?ver=1.4.4 HTTP/1.1" 200 1490 "http://intranet.price.fox.org/" "Mozilla
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-includes/js/wp-embed.min.js?ver=5.8.1 HTTP/1.1" 200 1099 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/53
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-includes/js/jquery/jquery-migrate.min.js?ver=3.3.2 HTTP/1.1" 200 4505 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x86_64) A
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-content/themes/go/dist/js/frontend.min.js?ver=1.4.4 HTTP/1.1" 200 11448 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x86_64)
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-content/themes/go/dist/css/style-shared.min.css?ver=1.4.4 HTTP/1.1" 200 23724 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x
10.35.34.144 - - [04/Oct/2021:07:17:02 +0000] "GET /wp-includes/js/jquery/jquery.min.js?ver=3.6.0 HTTP/1.1" 200 31246 "http://intranet.price.fox.org/" "Mozilla/5.0 (X11; Linux x86_64) AppleWeb
```

COMMON LOG FORMATS

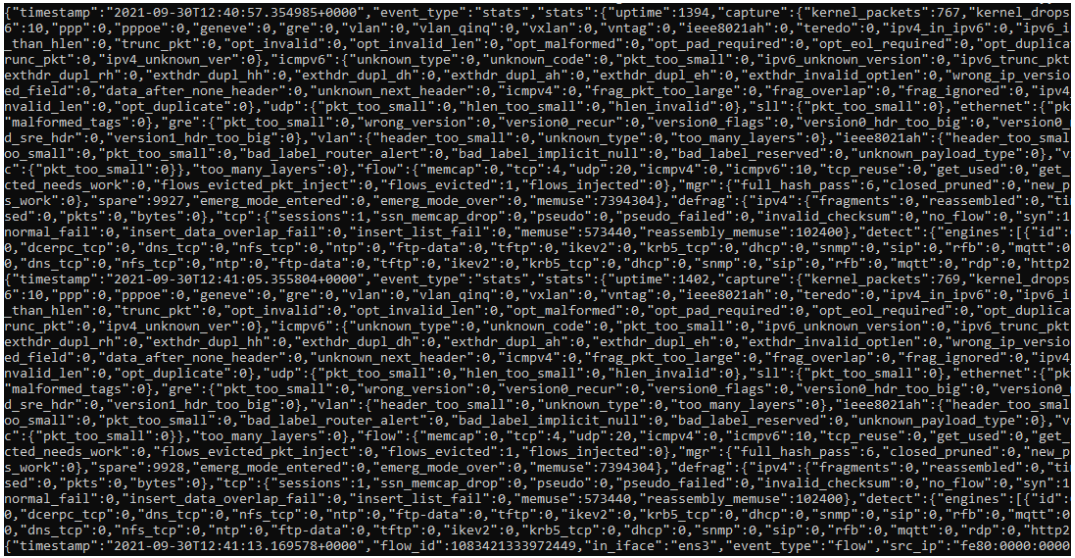
- **Audit logs**
  - Key-value pairs
  - High granularity
  - Encoded values

Access logs

```
type=USER_ACCT msg=audit(1633018621.978:197): pid=9709 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:accounting acct="root" exe="/usr/sbin/cron"
type=CRED_ACQ msg=audit(1633018621.982:198): pid=9709 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:setcred acct="root" exe="/usr/sbin/cron" host
type=LOGIN msg=audit(1633018621.982:199): pid=9709 uid=0 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=88 res=1
type=USER_START msg=audit(1633018621.982:200): pid=9709 uid=0 auid=0 ses=88 msg='op=PAM:session_open acct="root" exe="/usr/sbin/cron" hostname=? add
type=CRED_DISP msg=audit(1633018621.990:201): pid=9709 uid=0 auid=0 ses=88 msg='op=PAM:setcred acct="root" exe="/usr/sbin/cron" hostname=? addr=? te
type=USER_END msg=audit(1633018621.990:202): pid=9709 uid=0 auid=0 ses=88 msg='op=PAM:session_close acct="root" exe="/usr/sbin/cron" hostname=? addr
type=USER_ACCT msg=audit(1633019397.538:203): pid=9715 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:accounting acct="phopkins" exe="/usr/sbin/ss
type=CRED_ACQ msg=audit(1633019397.542:204): pid=9715 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:setcred acct="phopkins" exe="/usr/sbin/sshd"
type=LOGIN msg=audit(1633019397.542:205): pid=9715 uid=0 old-auid=4294967295 auid=1001 tty=(none) old-ses=4294967295 ses=89 res=1
type=USER_ACCT msg=audit(1633019397.578:206): pid=9717 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:accounting acct="phopkins" exe="/lib/systemd
type=CRED_ACQ msg=audit(1633019397.582:207): pid=9717 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:setcred acct="phopkins" exe="/lib/systemd/sys
type=LOGIN msg=audit(1633019397.582:208): pid=9717 uid=0 old-auid=4294967295 auid=1001 tty=(none) old-ses=4294967295 ses=90 res=1
type=USER_START msg=audit(1633019397.582:209): pid=9717 uid=0 auid=1001 ses=90 msg='op=PAM:session_open acct="phopkins" exe="/lib/systemd/systemd" h
type=SERVICE_START msg=audit(1633019397.646:210): pid=1 uid=0 auid=4294967295 ses=4294967295 msg='unit=user@1001 comm="systemd" exe="/lib/systemd/sy
type=USER_START msg=audit(1633019398.598:211): pid=9715 uid=0 auid=1001 ses=89 msg='op=PAM:session_open acct="phopkins" exe="/usr/sbin/sshd" hostnam
type=CRED_ACQ msg=audit(1633019398.602:212): pid=9831 uid=0 auid=1001 ses=89 msg='op=PAM:setcred acct="phopkins" exe="/usr/sbin/sshd" hostname=10.35
type=USER_LOGIN msg=audit(1633019398.646:213): pid=9715 uid=0 auid=1001 ses=89 msg='op=login id=1001 exe="/usr/sbin/sshd" hostname=10.35.35.27 addr=
type=USER_ACCT msg=audit(1633019941.018:214): pid=9851 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:accounting acct="root" exe="/usr/sbin/cron"
type=CRED_ACQ msg=audit(1633019941.022:215): pid=9851 uid=0 auid=4294967295 ses=4294967295 msg='op=PAM:setcred acct="root" exe="/usr/sbin/cron" host
type=LOGIN msg=audit(1633019941.022:216): pid=9851 uid=0 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=91 res=1
```

COMMON LOG FORMATS

- **Event logs**
  - Json format
  - Many attributes, high volume



Access logs

Audit logs

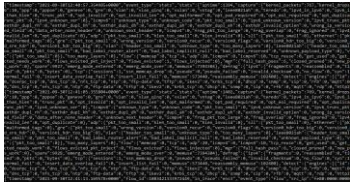
COMMON LOG FORMATS

- **System logs**
  - Unstructured (human readable messages)
  - Diverse event formats
  - Correlations (Starting <service> → Started <service>)

```
Oct  5 06:40:39 intranet-server systemd[538]: Received SIGRTMIN+24 from PID 633 (kill).
Oct  5 06:40:39 intranet-server systemd[1]: user@1000.service: Killing process 633 (kill) with signal SIGK
Oct  5 06:40:39 intranet-server systemd[1]: Stopped User Manager for UID 1000.
Oct  5 06:40:39 intranet-server systemd[1]: Removed slice User Slice of ait.
Oct  5 06:40:44 intranet-server systemd[1]: Created slice User Slice of ait.
Oct  5 06:40:44 intranet-server systemd[1]: Starting User Manager for UID 1000...
Oct  5 06:40:44 intranet-server systemd[1]: Started Session 454 of user ait.
Oct  5 06:40:44 intranet-server systemd[636]: Reached target Paths.
Oct  5 06:40:44 intranet-server systemd[636]: Listening on GnuPG network certificate management daemon.
Oct  5 06:40:44 intranet-server systemd[636]: Listening on GnuPG cryptographic agent (ssh-agent emulation)
Oct  5 06:40:44 intranet-server systemd[636]: Listening on REST API socket for snapd user session agent.
Oct  5 06:40:44 intranet-server systemd[636]: Listening on GnuPG cryptographic agent and passphrase cache
Oct  5 06:40:44 intranet-server systemd[636]: Reached target Timers.
Oct  5 06:40:44 intranet-server systemd[636]: Listening on GnuPG cryptographic agent and passphrase cache
Oct  5 06:40:44 intranet-server systemd[636]: Listening on GnuPG cryptographic agent and passphrase cache.
Oct  5 06:40:44 intranet-server systemd[636]: Reached target Sockets.
Oct  5 06:40:44 intranet-server systemd[636]: Reached target Basic System.
Oct  5 06:40:44 intranet-server systemd[1]: Started User Manager for UID 1000.
Oct  5 06:40:44 intranet-server systemd[636]: Reached target Default.
Oct  5 06:40:44 intranet-server systemd[636]: Startup finished in 66ms.
Oct  5 06:58:01 intranet-server systemd[1]: Starting Daily apt upgrade and clean activities...
Oct  5 06:58:18 intranet-server systemd[1]: Started Daily apt upgrade and clean activities.
Oct  5 07:01:11 intranet-server systemd[1]: Stopping User Manager for UID 1000...
Oct  5 07:01:11 intranet-server systemd[636]: Stopped target Default.
Oct  5 07:01:11 intranet-server systemd[636]: Stopped target Basic System.
Oct  5 07:01:11 intranet-server systemd[636]: Stopped target Paths.
Oct  5 07:01:12 intranet-server systemd[636]: Stopped target Sockets.
```

Access logs

Audit logs

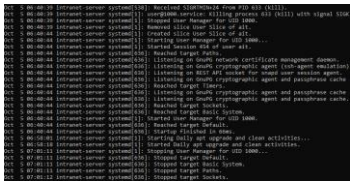


Event logs

COMMON LOG FORMATS

- **DNS logs**
  - User queries

```
Sep 29 12:12:41 dnsmasq[14715]: cached pypi.org is 2a04:4e42:600::223
Sep 29 12:12:41 dnsmasq[14715]: cached pypi.org is 2a04:4e42:200::223
Sep 29 12:12:41 dnsmasq[14715]: cached pypi.org is 2a04:4e42::223
Sep 29 12:12:41 dnsmasq[14715]: query[A] files.pythonhosted.org from 172.17.130.23
Sep 29 12:12:41 dnsmasq[14715]: cached files.pythonhosted.org is <CNAME>
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 151.101.129.63
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 151.101.193.63
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 151.101.1.63
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 151.101.65.63
Sep 29 12:12:41 dnsmasq[14715]: query[AAAA] files.pythonhosted.org from 172.17.130.23
Sep 29 12:12:41 dnsmasq[14715]: cached files.pythonhosted.org is <CNAME>
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 2a04:4e42:400::319
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 2a04:4e42:600::319
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 2a04:4e42::319
Sep 29 12:12:41 dnsmasq[14715]: cached dualstack.r.ssl.global.fastly.net is 2a04:4e42:200::319
Sep 29 12:12:46 dnsmasq[14715]: query[A] pypi.python.org from 172.17.128.1
Sep 29 12:12:46 dnsmasq[14715]: cached pypi.python.org is <CNAME>
Sep 29 12:12:46 dnsmasq[14715]: cached dualstack.python.map.fastly.net is 151.101.112.223
Sep 29 12:12:46 dnsmasq[14715]: query[AAAA] pypi.python.org from 172.17.128.1
Sep 29 12:12:46 dnsmasq[14715]: cached pypi.python.org is <CNAME>
Sep 29 12:12:46 dnsmasq[14715]: cached dualstack.python.map.fastly.net is 2a04:4e42:1b::223
Sep 29 12:12:46 dnsmasq[14715]: query[AAAA] dualstack.python.map.fastly.net from 172.17.128.1
Sep 29 12:12:46 dnsmasq[14715]: cached dualstack.python.map.fastly.net is 2a04:4e42:1b::223
Sep 29 12:12:46 dnsmasq[22613]: query[A] pypi.python.org from 10.35.32.1
Sep 29 12:12:46 dnsmasq[22613]: cached pypi.python.org is <CNAME>
```



System logs

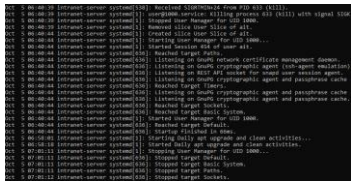
Access logs

Audit logs

Event logs

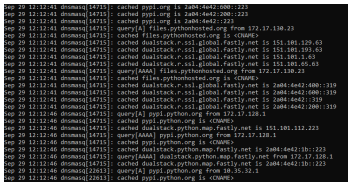
COMMON LOG FORMATS

- **Application logs**
  - E.g., horde mail logs
  - User behavior in specific context



System logs

Access logs



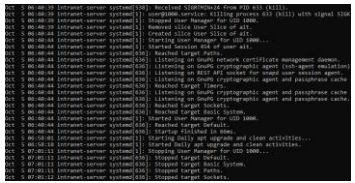
Audit logs

Event logs

```
Oct  4 09:09:53 jonesmorgan-mail HORDE: [imp] Login success for susan.gregory (192.168.129.210) to {imap://smtp.jonesmorgan.fox.org/} [pid 2154
Oct  4 09:11:39 jonesmorgan-mail HORDE: [imp] Message sent to meghan.morris@taylorcruz.fox.org from susan.gregory (192.168.129.210) [pid 21543
Oct  4 09:13:20 jonesmorgan-mail HORDE: [imp] Message sent to carole.noble@jonesmorgan.fox.org from susan.gregory (192.168.129.210) [pid 20533
Oct  4 09:15:06 jonesmorgan-mail HORDE: [horde] User susan.gregory logged out of Horde (192.168.129.210) [pid 20933 on line 106 of "/usr/share/h
Oct  4 09:31:25 jonesmorgan-mail HORDE: [horde] Login success for susan.gregory to horde (192.168.129.210) [pid 21782 on line 163 of "/usr/shar
Oct  4 09:31:25 jonesmorgan-mail HORDE: [imp] Login success for susan.gregory (192.168.129.210) to {imap://smtp.jonesmorgan.fox.org/} [pid 2178
Oct  4 09:34:55 jonesmorgan-mail HORDE: [imp] Message sent to Mail Delivery System <Mailer-Daemon@smtp.jonesmorgan.fox.org> from susan.gregory
Oct  4 09:36:37 jonesmorgan-mail HORDE: [imp] Message sent to daniel.morgan@taylorcruz.fox.org from susan.gregory (192.168.129.210) [pid 21905
Oct  4 09:40:30 jonesmorgan-mail HORDE: [imp] Message sent to meghan.morris@taylorcruz.fox.org, carole.noble@jonesmorgan.fox.org from susan.greg
Oct  4 09:41:49 jonesmorgan-mail HORDE: [imp] Message sent to patrick.hopkins@miller.fox.org from susan.gregory (192.168.129.210) [pid 21782 on
Oct  4 09:45:26 jonesmorgan-mail HORDE: [imp] Message sent to carole.noble@jonesmorgan.fox.org, damian.jordan@taylorcruz.fox.org from susan.greg
Oct  4 09:49:16 jonesmorgan-mail HORDE: [imp] Message sent to meghan.morris@taylorcruz.fox.org from susan.gregory (192.168.129.210) [pid 20933
Oct  4 09:51:04 jonesmorgan-mail HORDE: [imp] Message sent to Mail Delivery System <Mailer-Daemon@smtp.jonesmorgan.fox.org> from susan.gregory
Oct  4 09:52:23 jonesmorgan-mail HORDE: [imp] Message sent to patrick.hopkins@miller.fox.org from susan.gregory (192.168.129.210) [pid 21900 on
Oct  4 09:55:44 jonesmorgan-mail HORDE: [horde] User susan.gregory logged out of Horde (192.168.129.210) [pid 20954 on line 106 of "/usr/share/h
Oct  4 10:39:14 jonesmorgan-mail HORDE: [horde] Login success for susan.gregory to horde (192.168.129.210) [pid 20933 on line 163 of "/usr/shar
Oct  4 10:39:14 jonesmorgan-mail HORDE: [imp] Login success for susan.gregory (192.168.129.210) to {imap://smtp.jonesmorgan.fox.org/} [pid 2093
Oct  4 10:41:55 jonesmorgan-mail HORDE: [imp] Message sent to jessica.wright@whittaker.fox.org from susan.gregory (192.168.129.210) [pid 20933
Oct  4 10:44:07 jonesmorgan-mail HORDE: [imp] Message sent to samantha.thornton@whittaker.fox.org from susan.gregory (192.168.129.210) [pid 217
Oct  4 10:50:19 jonesmorgan-mail HORDE: [imp] Message sent to daniel.morgan@taylorcruz.fox.org from susan.gregory (192.168.129.210) [pid 21782
```

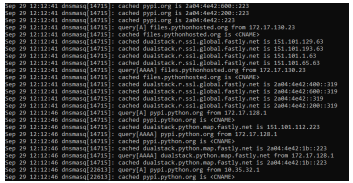
COMMON LOG FORMATS

- Many formats
  - No common standards
- Diverse events
  - Creating parsers can be tough!
  - No well-defined syntax
  - No complete list of events available
- Two main goals of log parsing:
  - Differentiate events
  - Access specific tokens



System logs

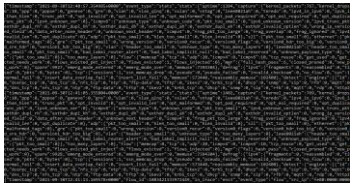
Access logs



DNS logs

Audit logs

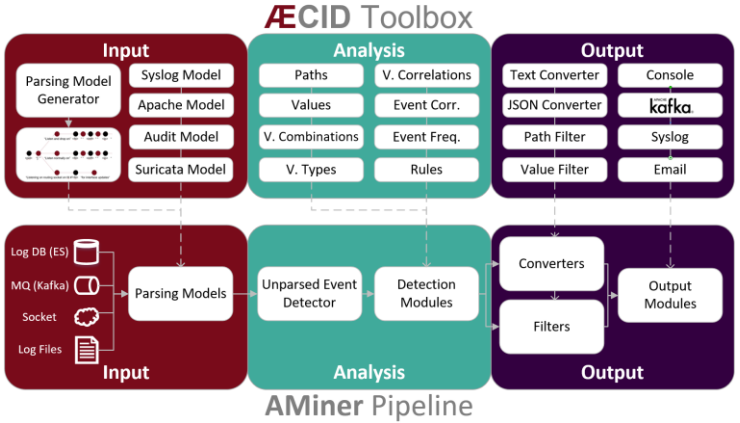
Application logs



Event logs

INTRODUCING ÆCID AND AMINER

- **AECID** is a mature intrusion detection system using computer log data
  - Ingests **log data from any system**
  - Works with domain specific and previously unknown systems, i.e., does not rely on predefined parsers – **self-learning**!
  - **Light-weight**, distributed anomaly detection
  - Clients run with **low memory footprint** and minimum CPU utilization
  - **Not in competition** with well-established systems, but as **additional detection mechanism**
    - Proof-of-Concept deployments as **sensors** for **ELK Stack** and **QRadar SIEMs**



https://github.com/ait-aecid/logdata-anomaly-miner

THE ÆCID APPROACH

1. **Log parser** generation
   - A “recipe” on how to dissect log lines of unknown grammar
   - Make log data usable for analysis → structured representation & easy to access
2. **Hypotheses** proposal
   - Distribution of property values (e.g., IP addresses, user names, …) in single events
   - And across multiple events
   - Correlation of event types
3. **Rule** generation through continuous hypotheses evaluation
   - Sort out unstable hypotheses and create rules for stable ones
   - Constitution of the system behavior model (learned behavior model)
4. **Anomaly detection**: rate the deviation of actual system behavior from the learned behavior model (anomalous points / context / frequency / sequence of events)

All steps take place in parallel, i.e., even during the anomaly detection phase, new hypotheses are created on the fly.

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

PART I:
FLEXIBLE CREATION OF LOG DATA PARSERS

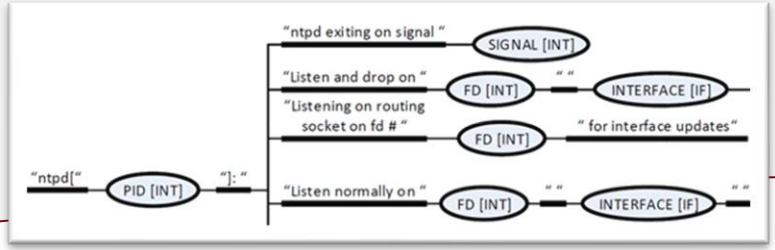
FAST LOG DATA PARSING

- Using **regex** for parsing is **inefficient** → **O(n)**
- Represent log line model as **tree-like graph** → **O(log(n))** → **Parse data once!**
  - Describes information most efficiently – with **minimal storage requirements**
  - **Efficient** log line **processing, classification** and **information access**

How to generate an efficient tree-like log parser?

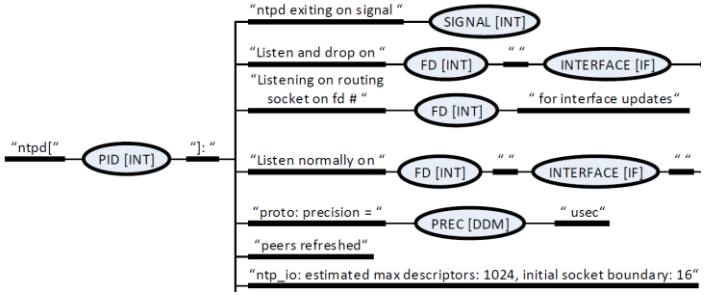
```
Oct 15 00:10:27 mymachine ntpd[16721]: Listen normally on 2 lo  127.0.0.1 UDP 123
```

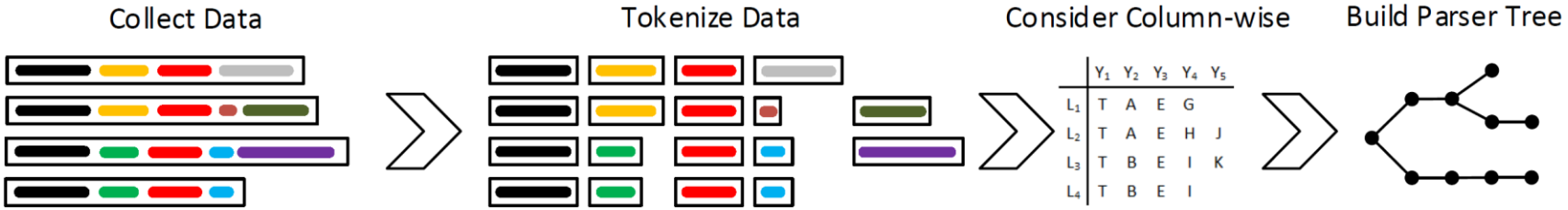
/model/syslog/time: 2021-10-15 00:10:27
/model/syslog/host: mymachine
/model/services/ntpd/pid: ntpd
/model/services/ntpd/listen/fd: 2
/model/services/ntpd/listen/if: 127.0.0.1



AECID-PG – CONCEPT

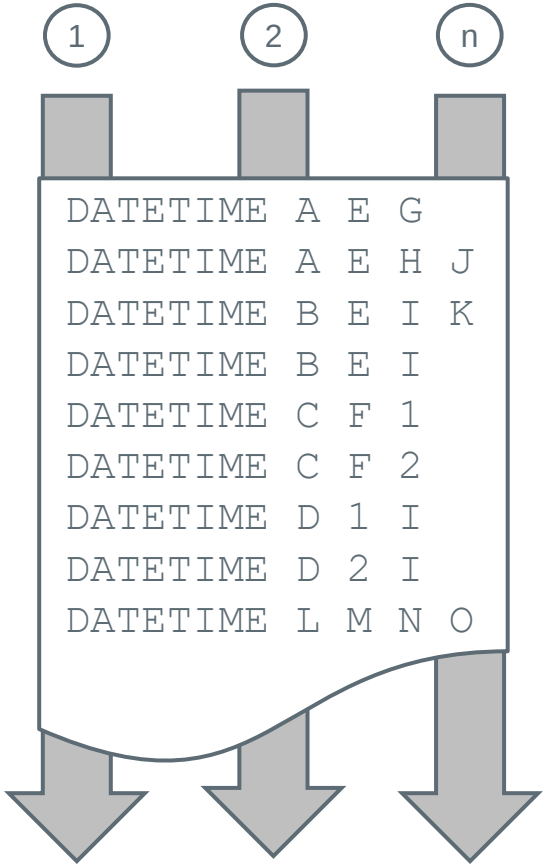
- Density based approach
- Independent from semantics
- Detect static and variable log line parts
- Build model from a **local point of view**
  - Different log line classes should not influence each other
- **Prohibit overfitting**

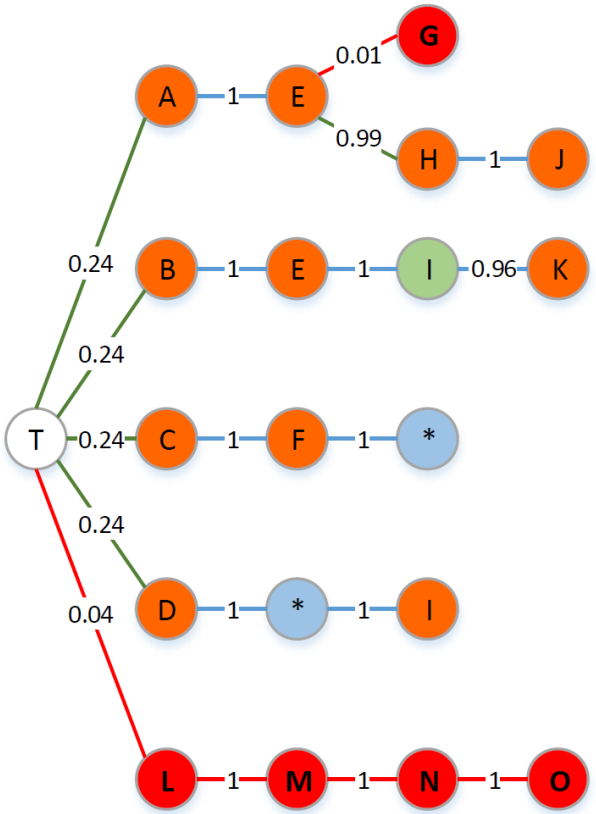




AECID-PG: PARSER GENERATOR

- Nodes:
  - Static
  - Variable
  - Optional
  - Deleted





SAMPLE LOGS

ntpd [16721]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
ntpd [16721]: Listen and drop on 1 v6wildcard :: UDP 123
ntpd [16721]: Listen normally on 2 lo 127.0.0.1 UDP 123
ntpd [16721]: Listen normally on 3 eth0 134.74.77.21 UDP 123
ntpd [16721]: Listen normally on 4 eth1 10.10.0.57 UDP 123
ntpd [16721]: Listen normally on 5 eth1 fe80::5652:ff:fe5a:f89f UDP 123
ntpd [16721]: Listen normally on 6 eth0 fe80::5652:ff:fe01:1fff UDP 123
ntpd [16721]: Listening on routing socket on fd #24 for interface updates

LOG TEMPLATES

ntpd [16721]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
ntpd [16721]: Listen and drop on 1 v6wildcard :: UDP 123
ntpd [16721]: Listen normally on 2 lo 127.0.0.1 UDP 123
ntpd [16721]: Listen normally on 3 eth0 134.74.77.21 UDP 123
ntpd [16721]: Listen normally on 4 eth1 10.10.0.57 UDP 123
ntpd [16721]: Listen normally on 5 eth1 fe80::5652:ff:fe5a:f89f UDP 123
ntpd [16721]: Listen normally on 6 eth0 fe80::5652:ff:fe01:1fff UDP 123
ntpd [16721]: Listening on routing socket on fd #24 for interface updates

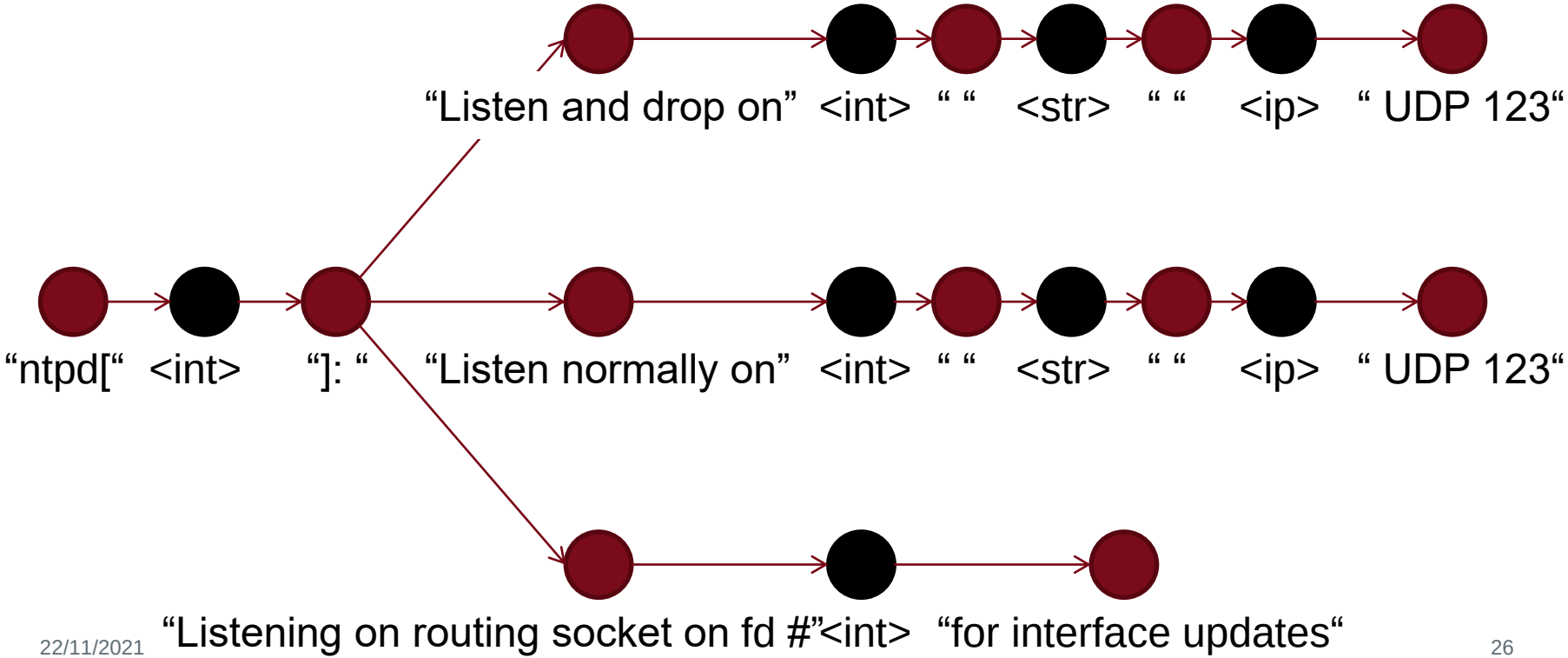
ntpd[§]: Listen and drop on § § § UDP 123
ntpd[§]: Listen normally on § § § UDP 123
ntpd[§]: Listening on routing socket on fd #§ for interface updates

LOG TEMPLATES

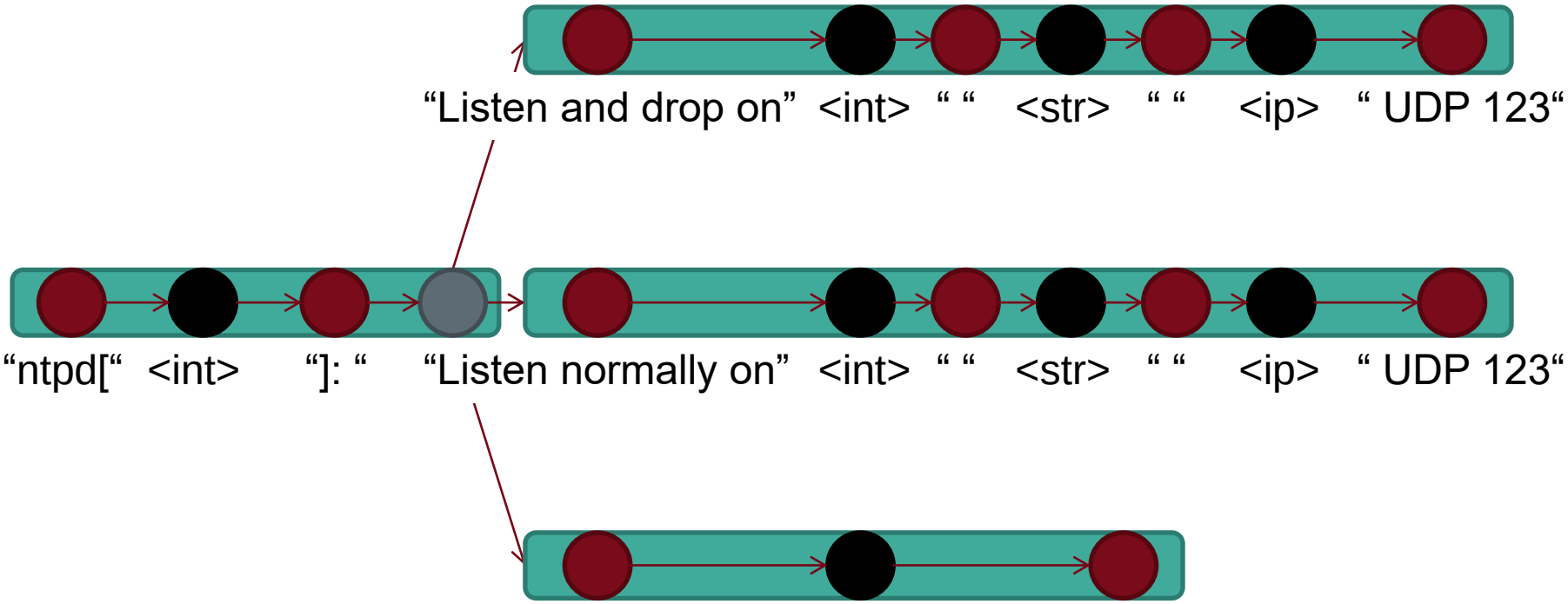
ntpd [16721]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
ntpd [16721]: Listen and drop on 1 v6wildcard :: UDP 123
ntpd [16721]: Listen normally on 2 lo 127.0.0.1 UDP 123
ntpd [16721]: Listen normally on 3 eth0 134.74.77.21 UDP 123
ntpd [16721]: Listen normally on 4 eth1 10.10.0.57 UDP 123
ntpd [16721]: Listen normally on 5 eth1 fe80::5652:ff:fe5a:f89f UDP 123
ntpd [16721]: Listen normally on 6 eth0 fe80::5652:ff:fe01:1fff UDP 123
ntpd [16721]: Listening on routing socket on fd #24 for interface updates

ntpd[<int>]: Listen and drop on <int> <str> <ip> UDP 123
ntpd[<int>]: Listen normally on <int> <str> <ip> UDP 123
ntpd[<int>]: Listening on routing socket on fd #<int> for interface updates

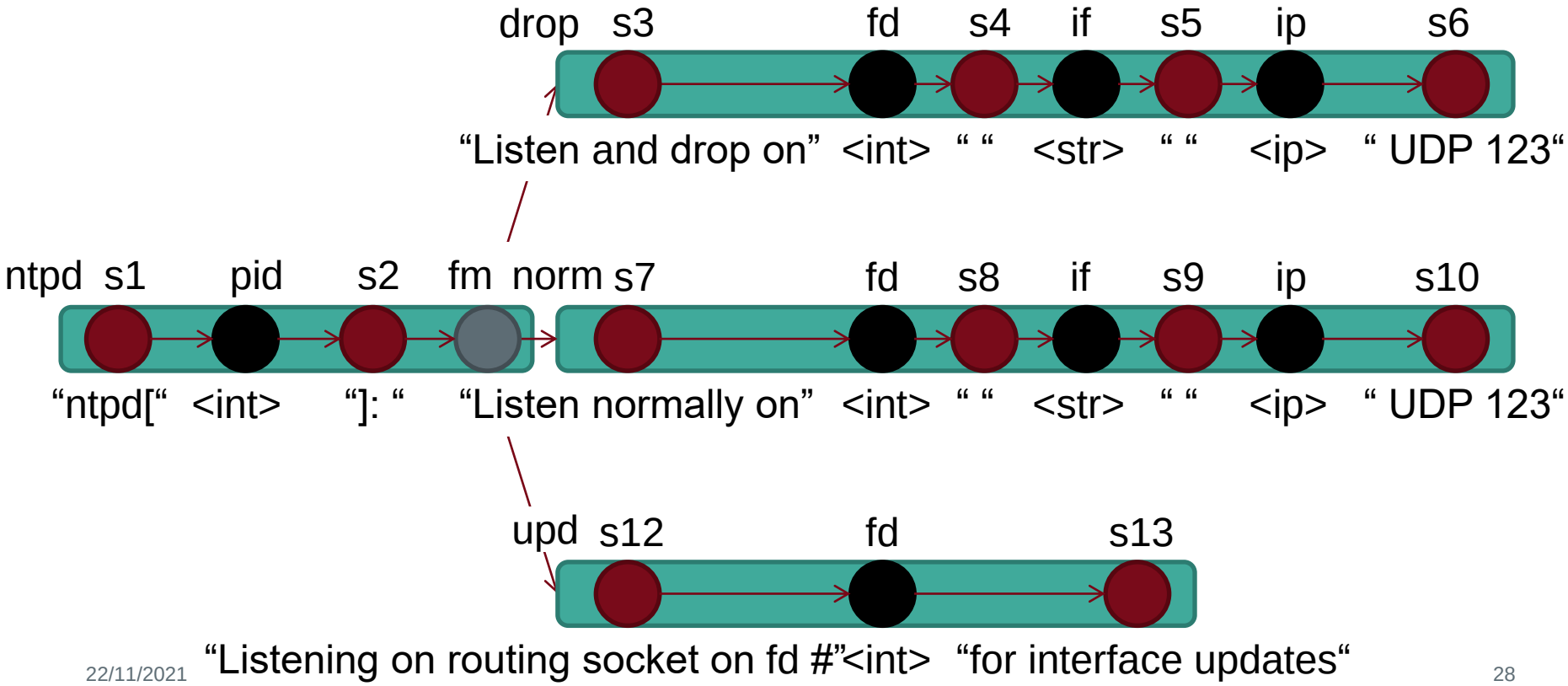
PARSER TREE


“Listen and drop on” <int> “ “ <str> “ “ <ip> “ UDP 123“
“ntpd[“ <int> “]: “
“Listen normally on” <int> “ “ <str> “ “ <ip> “ UDP 123“
“Listening on routing socket on fd #”<int> “for interface updates“

PARSER TREE

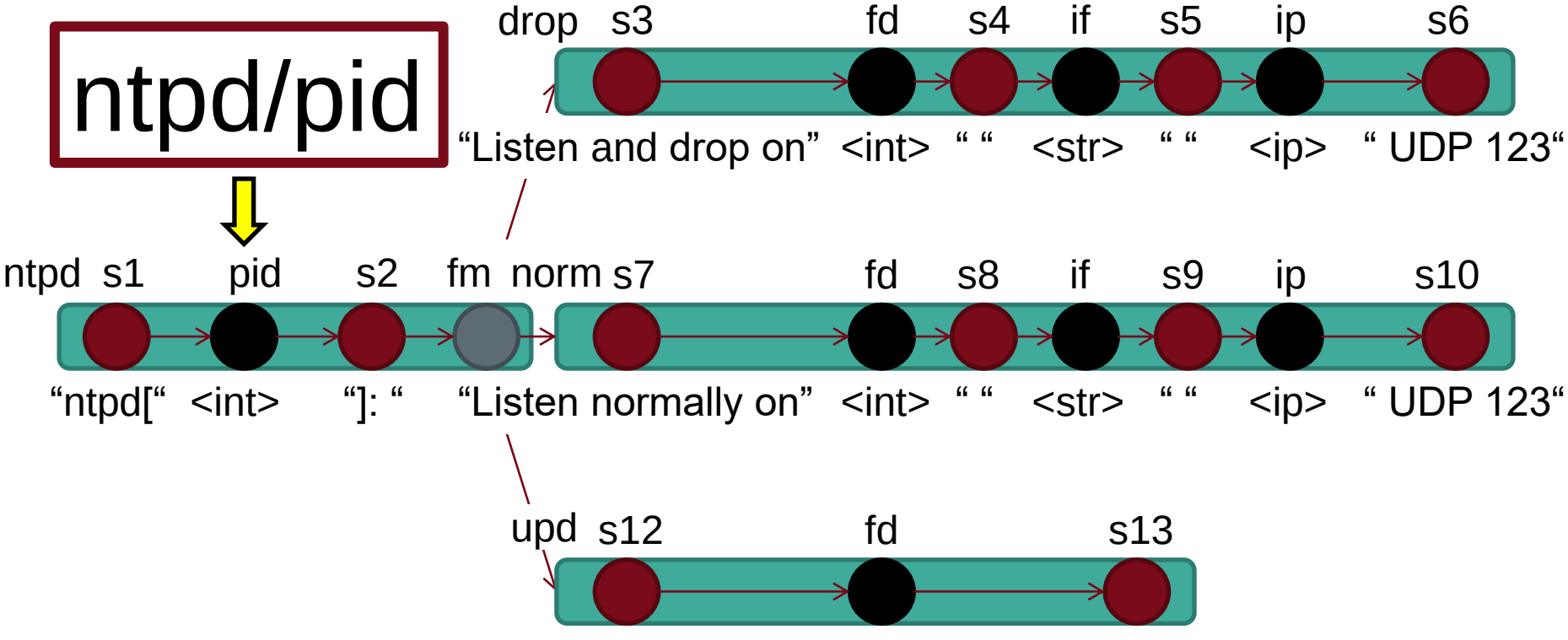

“Listen and drop on” <int> “ “ <str> “ “ <ip> “ UDP 123“
“ntpd[“ <int> “]: “ “Listen normally on” <int> “ “ <str> “ “ <ip> “ UDP 123“
“Listening on routing socket on fd #”<int> “for interface updates“

PARSER TREE

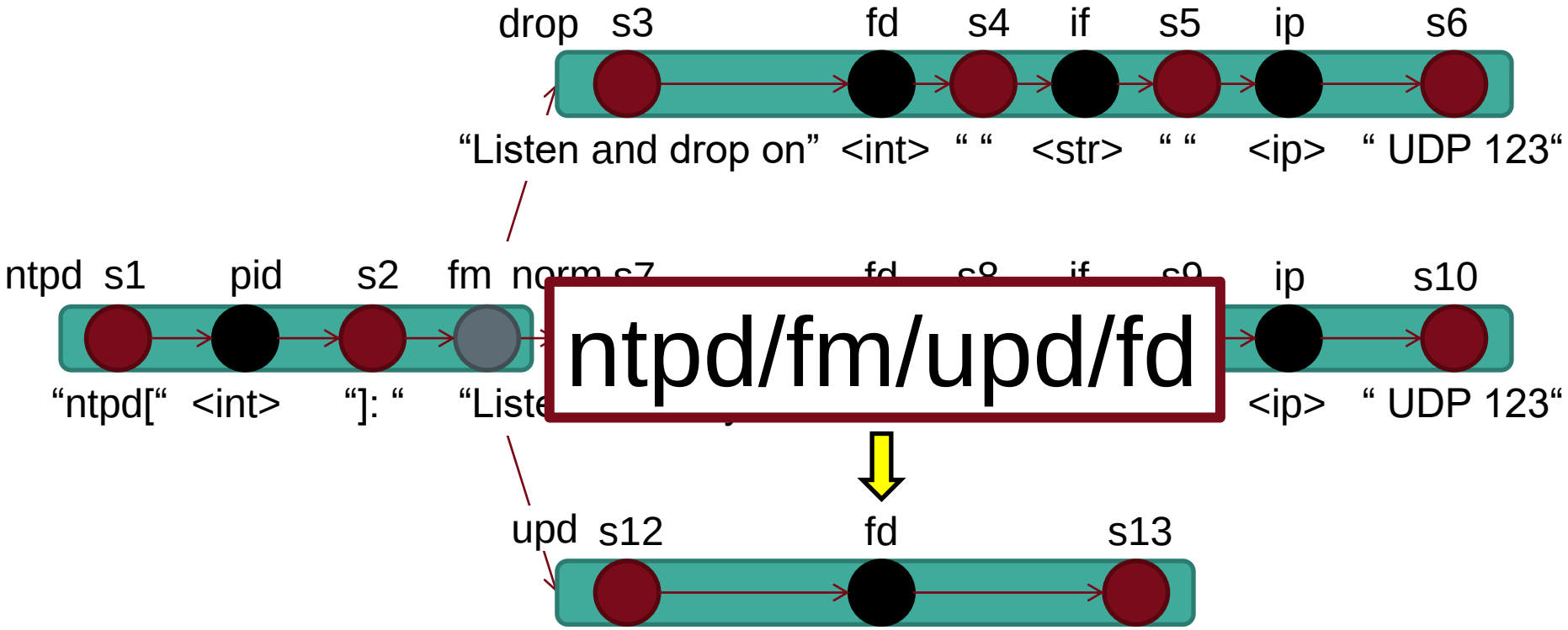

drop s3 fd s4 if s5 ip s6
“Listen and drop on” <int> “ “ <str> “ “ <ip> “ UDP 123“
ntpd s1 pid s2 fm norm s7 fd s8 if s9 ip s10
“ntpd[“ <int> “]: “ “Listen normally on” <int> “ “ <str> “ “ <ip> “ UDP 123“
upd s12 fd s13
“Listening on routing socket on fd #”<int> “for interface updates“

PARSER TREE

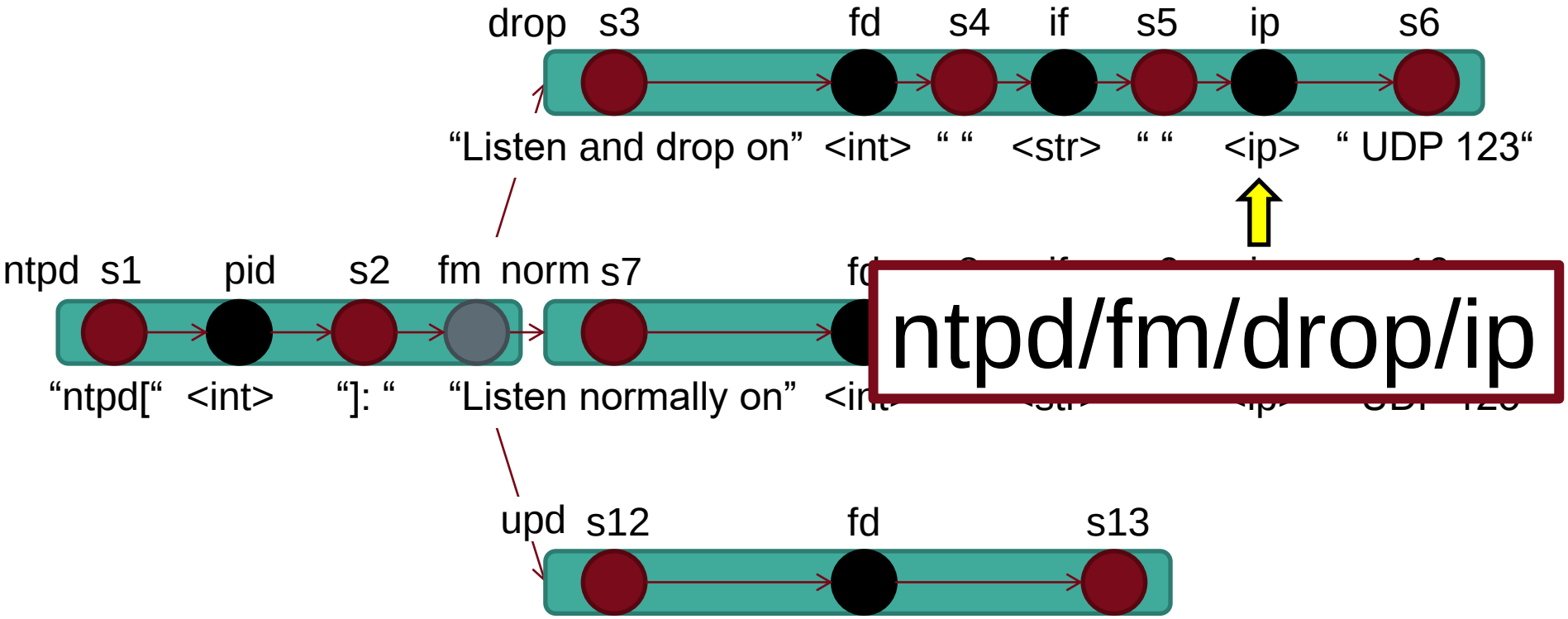
ntpd/pid



PARSER TREE


drop
s3
fd
s4
if
s5
ip
s6
“Listen and drop on”
<int>
“ “
<str>
“ “
<ip>
“ UDP 123“
ntpd
s1
pid
s2
fm
ip
s10
ntpd/fm/upd/fd
“ntpd[“
<int>
“]: “
<ip>
“ UDP 123“
upd
s12
fd
s13
“Listening on routing socket on fd #”<int>
“for interface updates“

PARSER TREE


drop s3 fd s4 if s5 ip s6
“Listen and drop on” <int> “ “ <str> “ “ <ip> “ UDP 123“
ntpd s1 pid s2 fm norm s7
“ntpd[“ <int> “]: “ “Listen normally on” <int>
ntpd/fm/drop/ip
upd s12 fd s13
“Listening on routing socket on fd #”<int> “for interface updates“

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

PARSER-GENERATOR

Live Demo

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

PART II:
ONLINE ANOMALY DETECTION

HYPOTHESES PROPOSAL AND ANOMALY DETECTION

```
Jun 20 00:59:37 localhost sshd[1008]: Accepted public key for backup
from 172.29.147.33 port 54149 ssh2: RSA SHA256:9k...
```

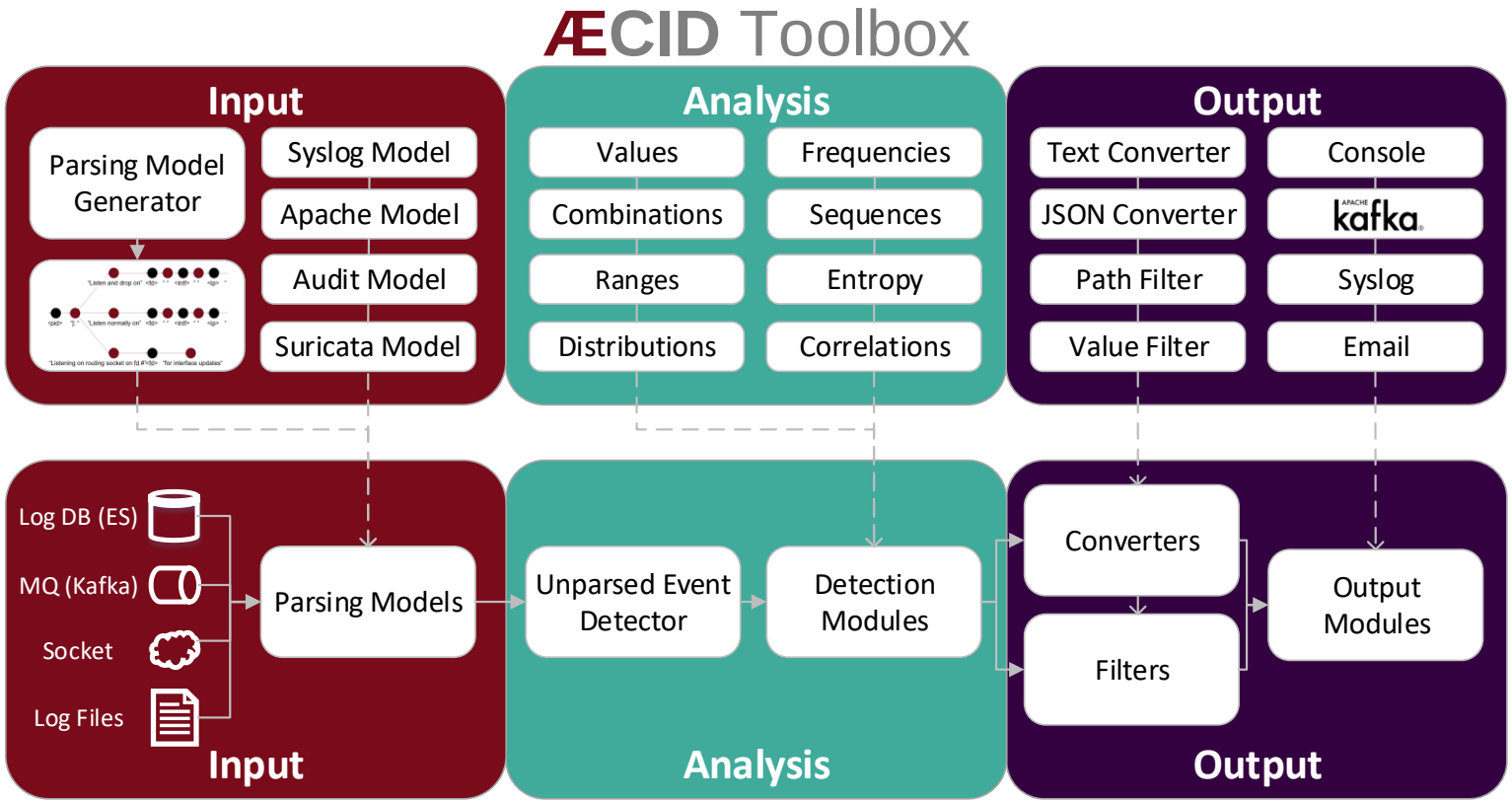
/model/syslog/time: **Jun 20 00:59:37**
/model/syslog/host: **localhost**
/model/services/sshd/sname: **sshd**
/model/services/sshd/msg/acceptedpk/pid: **1008**
/model/services/sshd/msg/acceptedpk/user: **backup**
/model/services/sshd/msg/acceptedpk/originip: **172.29.147.33**
/model/services/sshd/msg/acceptedpk/port: **54149**
/model/services/sshd/msg/acceptedpk/protocol: **ssh2**
/model/services/sshd/msg/acceptedpk/crypto: **RSA**
/model/services/sshd/msg/acceptedpk/fingerprint: **SHA256:9k...**

Simple Example Hypotheses:

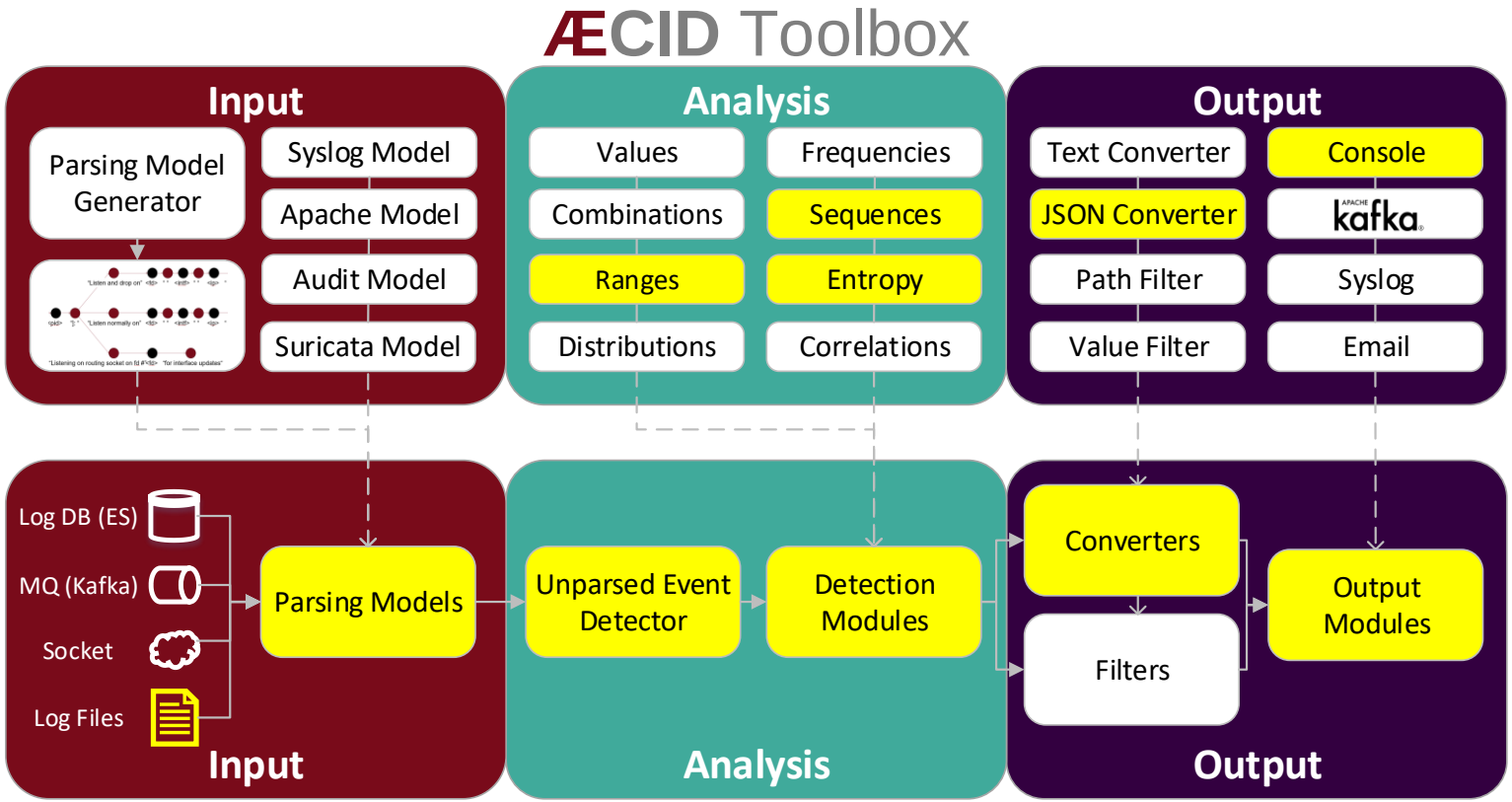
```
user{backup} ~ remoteip{172.29.147.33}
user{backup} ~ fingerprint{SHA256:9k...}
user{backup} only allowed in time_hh{[00,03]}
...
```

- Different Methods for hypothesis generation (incl. brute force)
- Coverage of events is complex to determine
- Maximize detection capabilities with minimum number of (stable) hypotheses
- Continuous learning in parallel to detection

AMINER PIPELINE



AMINER PIPELINE



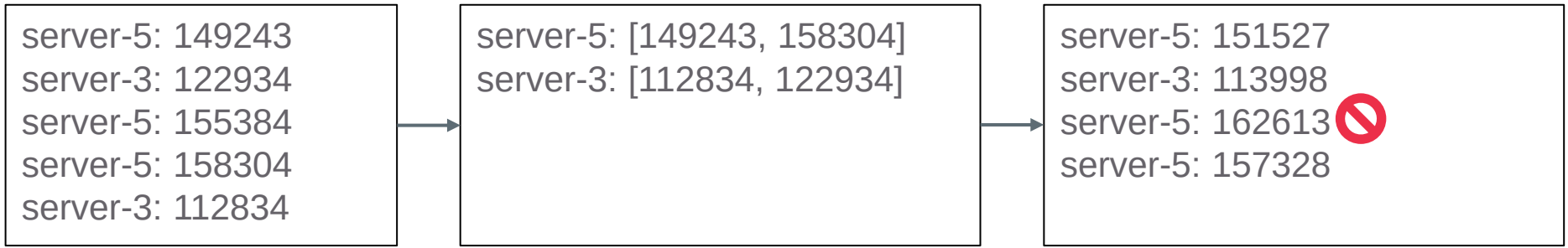
AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

AMINER

Live Demo

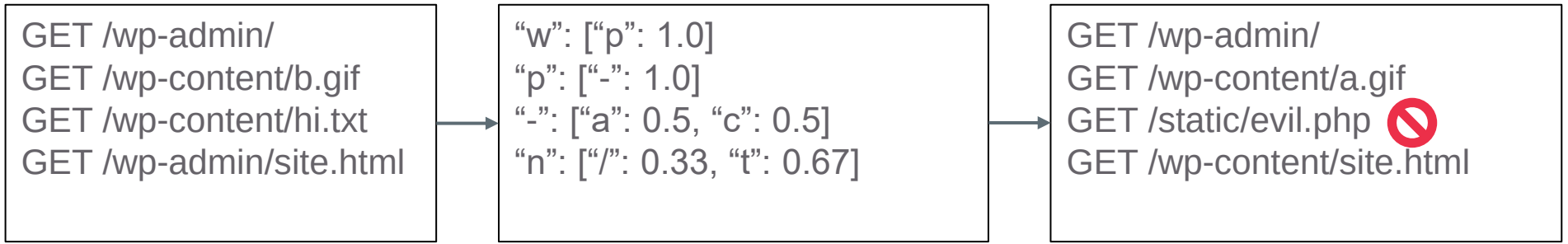
SCENARIO: OPERATIONAL TECHNOLOGY (OT)

- **System:** Hosts report system monitoring data: Temperature, bytes, syscalls
- **Data:** 17:05:51 server-5 systemd[1644]: Bytes sent: 162613
- **Attack:** Attacker installs crypto-miner on one of the hosts
- **Consequences:** Changes in monitoring data
- **Detection:** Value Range Detector
  - Learn minimum and maximum
  - Automatically expand and raise anomalies



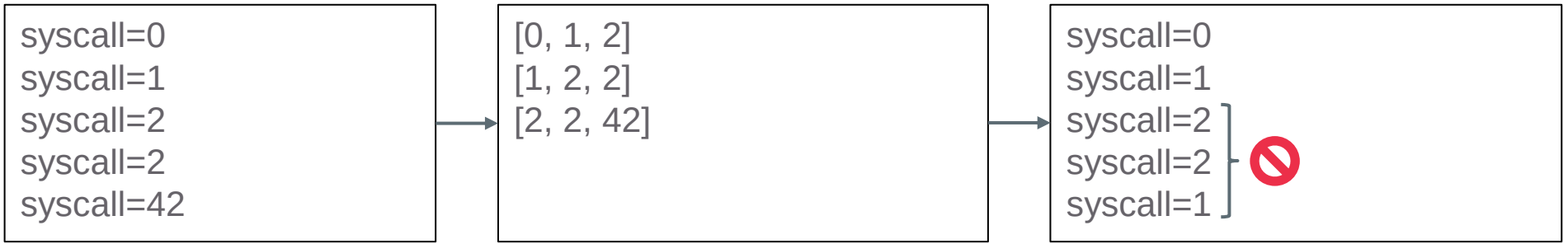
SCENARIO: REMOTE COMMAND EXEC

- **System:** Webserver with apache access logs
- **Data:** 10.35.34.9 - - [18:33:55] "GET /wp-admin/ HTTP/1.1" 302 361 "-" "Firefox/86.0"
- **Attack:** Webshell on server allows attacker to execute commands
- **Consequences:** Commands visible in accessed resources
- **Detection:** Entropy Detector
  - Learn probability distributions of character pairs, e.g., “/wp-admin/”, “/wp-content/”
  - Automatically update probabilities during training or use default table



SCENARIO: PROCESS HIJACKING

- **System:** Host running audit daemon
- **Data:** type=SYSCALL msg=audit(1583016732.264:4786292): syscall=0 success=yes
- **Attack:** Attacker modifies a process as part of a privilege escalation
- **Consequences:** Process carries out operations in different order than before
- **Detection:** Sequence Detector
  - Learns sequences of fixed lengths
  - Need to untangle interleaved processes



KEY TAKEAWAYS

- **Log data analysis** is a non-trivial task
  - Many different formats and events
  - High volume data requires efficient code
  - Live monitoring and learning needs incremental algorithms
- **AECID/AMiner** allows to …
  - … manually or semi-automatically generate parsers
  - … automatically train models in a semi-supervised manner
  - … forensically analyze log data sets
  - … detect anomalies in logs as soon as they occur

LINKS

- **AECID:** https://aecid.ait.ac.at/
- **AMiner** (Github)**:** https://github.com/ait-aecid/logdata-anomaly-miner
  - **Tutorials:** https://github.com/ait-aecid/logdata-anomaly-miner/wiki
- **AMiner** (Debian)**:** https://packages.debian.org/sid/misc/logdata-anomaly-miner
- **Publications + Patents:** https://aecid.ait.ac.at/further-information/
- **Current Projects:**

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

THANK YOU!

aecid@ait.ac.at