Hardening

Applied Information Security
Lecture 4

Recap: Foreknowledge

Attackers:

- Mindset, Phases

What can be accomplished by an attack:

- Code Injection
  Dynamic Evaluation, Insecure Deserialization, XSS, Command Inject, SQL Injection, Buffer Overflow, ...
- Side Channels
  Hardware, Network, Physical World (airgap), ...
- Social Engineering
  (spear)phishing, weapons of influence, dumpster diving, ...

Today’s Topics

- detect vulnerabilities
  - known
    - vulnerability scan & exploit
  - unknown
    - automatic, manual
- detect attacks
  - ongoing
    - intrusion detection
  - past
    - auditing, malware removal
- administration
  - hardening, firewall, isolation

Detect Vulnerabilities: Known

Known vulnerabilities;

How to find them?

How to exploit them?

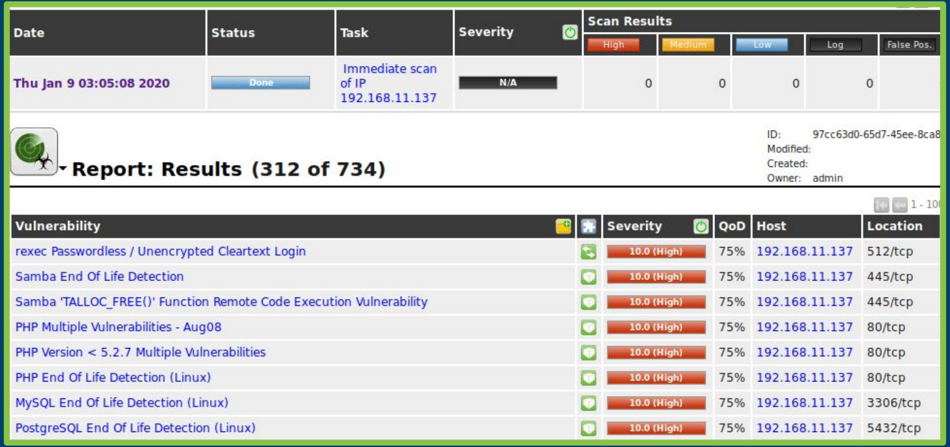
recently changed names; some still refer to this as **OpenVAS**

gvm

Greenbone Vulnerability Manager

MITRE CVE, + UI & auto-scan

- scan a specified host for vulnerabilities



Date	Status	Task	Severity	Scan Results				
				High	Medium	Low	Log	False Pos.
Thu Jan 9 03:05:08 2020	Done	Immediate scan of IP 192.168.11.137	N/A	0	0	0	0	

Report: Results (312 of 734)

ID: 97cc63d0-65d7-45ee-8ca8
Modified:
Created:
Owner: admin

1 - 10

Vulnerability	Severity	QoD	Host	Location
rexec Passwordless / Unencrypted Cleartext Login	10.0 (High)	75%	192.168.11.137	512/tcp
Samba End Of Life Detection	10.0 (High)	75%	192.168.11.137	445/tcp
Samba 'TALLOC_FREE()' Function Remote Code Execution Vulnerability	10.0 (High)	75%	192.168.11.137	445/tcp
PHP Multiple Vulnerabilities - Aug08	10.0 (High)	75%	192.168.11.137	80/tcp
PHP Version < 5.2.7 Multiple Vulnerabilities	10.0 (High)	75%	192.168.11.137	80/tcp
PHP End Of Life Detection (Linux)	10.0 (High)	75%	192.168.11.137	80/tcp
MySQL End Of Life Detection (Linux)	10.0 (High)	75%	192.168.11.137	3306/tcp
PostgreSQL End Of Life Detection (Linux)	10.0 (High)	75%	192.168.11.137	5432/tcp

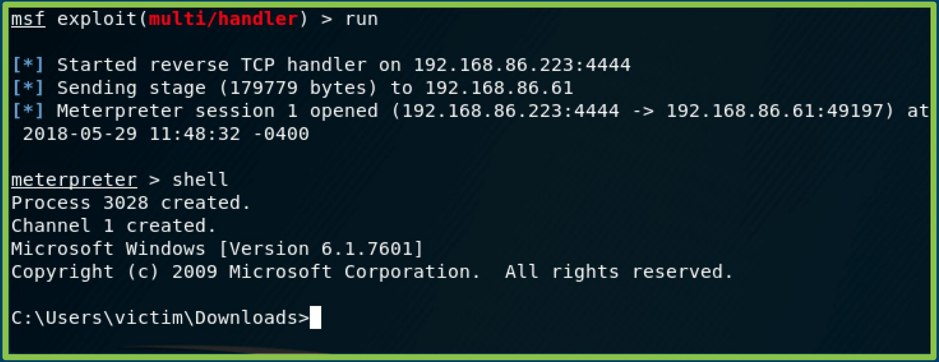
msfconsole

Metasploit Framework Console

collection of exploits

- standard format (use, options, run)
- Parameterized (IP address, ...)

for security auditing



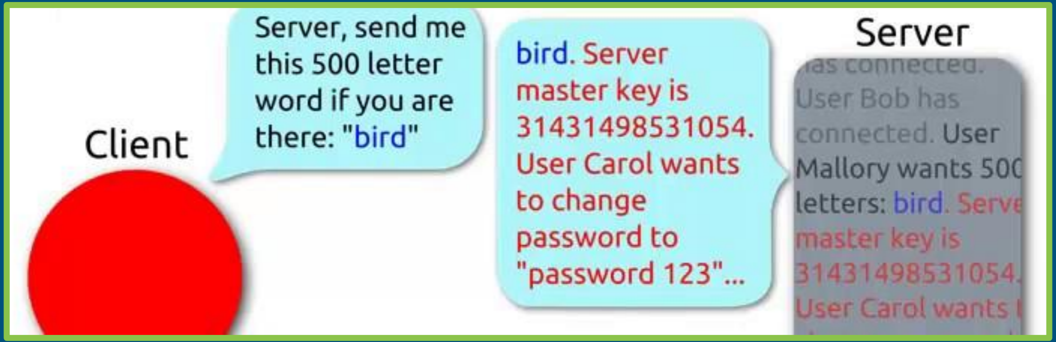
```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.86.223:4444
[*] Sending stage (179779 bytes) to 192.168.86.61
[*] Meterpreter session 1 opened (192.168.86.223:4444 -> 192.168.86.61:49197) at
 2018-05-29 11:48:32 -0400

meterpreter > shell
Process 3028 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\victim\Downloads>
```

Detect Vulnerabilities: Unknown

No known vulnerabilities; How do we craft new ones?

- automatic
- manual

Automatic


Client
Server, send me this 500 letter word if you are there: "bird"
bird. Server master key is 31431498531054. User Carol wants to change password to "password 123"...
Server
has connected. User Bob has connected. User Mallory wants 500 letters: bird. Serve master key is 31431498531054. User Carol wants

afl, sqlmap

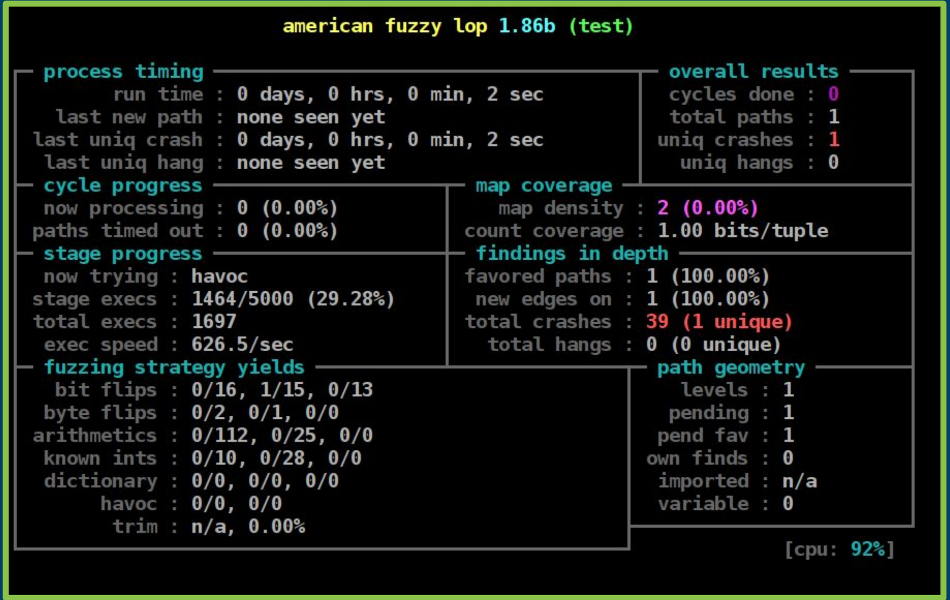
afl

American Fuzzy Lop

fuzzer

- randomly tweak a given input
- provoke bad behavior

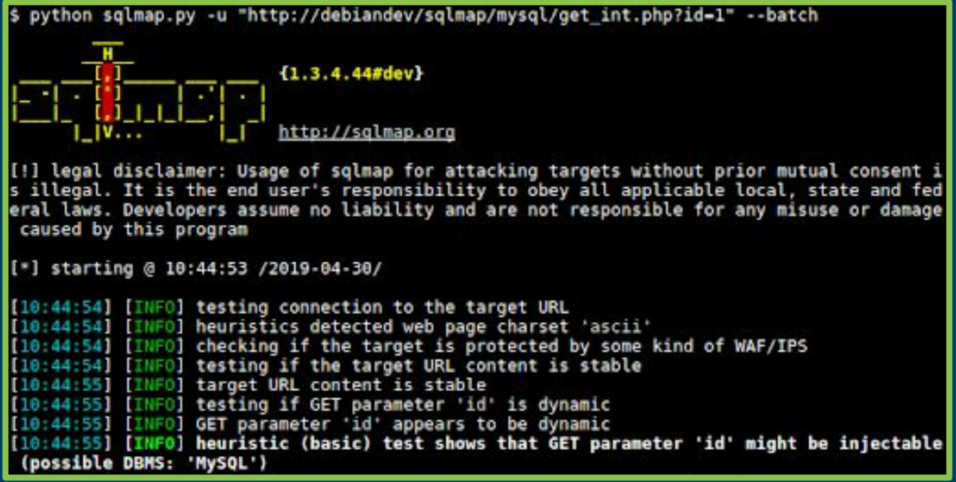
used e.g. to find buffer overflows.



```
                    american fuzzy lop 1.86b (test)

┌─ process timing ─────────────────────────────────────┬─ overall results ─────┐
│        run time : 0 days, 0 hrs, 0 min, 2 sec        │  cycles done : 0      │
│   last new path : none seen yet                      │  total paths : 1      │
│ last uniq crash : 0 days, 0 hrs, 0 min, 2 sec        │ uniq crashes : 1      │
│  last uniq hang : none seen yet                      │   uniq hangs : 0      │
├─ cycle progress ────────────────────┬─ map coverage ─┴───────────────────────┤
│  now processing : 0 (0.00%)         │    map density : 2 (0.00%)             │
│ paths timed out : 0 (0.00%)         │ count coverage : 1.00 bits/tuple       │
├─ stage progress ────────────────────┼─ findings in depth ────────────────────┤
│  now trying : havoc                 │ favored paths : 1 (100.00%)            │
│ stage execs : 1464/5000 (29.28%)    │  new edges on : 1 (100.00%)            │
│ total execs : 1697                  │ total crashes : 39 (1 unique)          │
│  exec speed : 626.5/sec             │   total hangs : 0 (0 unique)           │
├─ fuzzing strategy yields ───────────┴───────────┬─ path geometry ────────────┤
│   bit flips : 0/16, 1/15, 0/13                  │    levels : 1              │
│  byte flips : 0/2, 0/1, 0/0                     │   pending : 1              │
│ arithmetics : 0/112, 0/25, 0/0                  │  pend fav : 1              │
│  known ints : 0/10, 0/28, 0/0                   │ own finds : 0              │
│  dictionary : 0/0, 0/0, 0/0                     │  imported : n/a            │
│       havoc : 0/0, 0/0                          │  variable : 0              │
│        trim : n/a, 0.00%                        ├────────────────────────────┘
└─────────────────────────────────────────────────┘             [cpu: 92%]
```

sqlmap

test for SQL injections!



```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4.44#dev}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

Manual

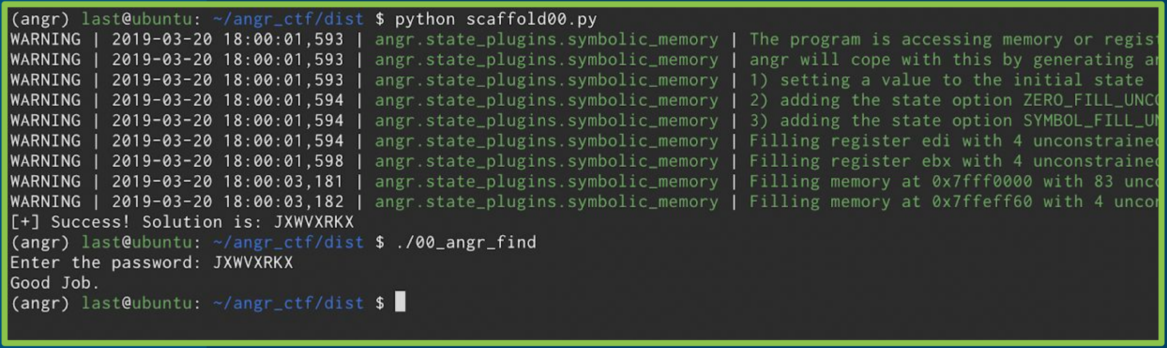
angr, Ghidra (reverse engineer), Burpsuite, ...

angr

binary analysis: concolic execution

- symbolic execution
  (abstract execution traces)
- + testing
  (to enter each abstract trace)

used e.g. to find buffer overflows.



```
(angr) last@ubuntu: ~/angr_ctf/dist $ python scaffold00.py
WARNING | 2019-03-20 18:00:01,593 | angr.state_plugins.symbolic_memory | The program is accessing memory or regist
WARNING | 2019-03-20 18:00:01,593 | angr.state_plugins.symbolic_memory | angr will cope with this by generating ar
WARNING | 2019-03-20 18:00:01,593 | angr.state_plugins.symbolic_memory | 1) setting a value to the initial state
WARNING | 2019-03-20 18:00:01,594 | angr.state_plugins.symbolic_memory | 2) adding the state option ZERO_FILL_UNCO
WARNING | 2019-03-20 18:00:01,594 | angr.state_plugins.symbolic_memory | 3) adding the state option SYMBOL_FILL_UN
WARNING | 2019-03-20 18:00:01,594 | angr.state_plugins.symbolic_memory | Filling register edi with 4 unconstrained
WARNING | 2019-03-20 18:00:01,598 | angr.state_plugins.symbolic_memory | Filling register ebx with 4 unconstrained
WARNING | 2019-03-20 18:00:03,181 | angr.state_plugins.symbolic_memory | Filling memory at 0x7fff0000 with 83 unco
WARNING | 2019-03-20 18:00:03,182 | angr.state_plugins.symbolic_memory | Filling memory at 0x7ffeff60 with 4 uncor
[+] Success! Solution is: JXWVXRKX
(angr) last@ubuntu: ~/angr_ctf/dist $ ./00_angr_find
Enter the password: JXWVXRKX
Good Job.
(angr) last@ubuntu: ~/angr_ctf/dist $ █
```

Detect Attacks: Ongoing

Attack is taking place!

How will we know?

How will we handle it?

- Intrusion detection

Intrusion Detection

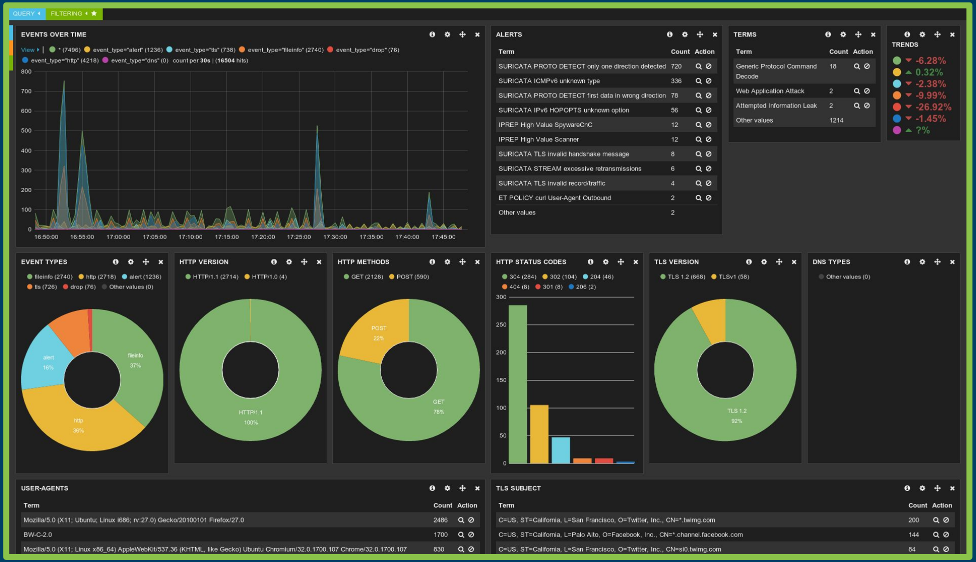
Intrusion Detection System

- automated review and response
- responds in (nearly) real time
- components:
  - sensors
  - analysis engine
  - countermeasure deployment
  - audit log

Intrusion Detection - Intrusion Detection System

Example: Network Monitoring

- suspicious behavior:
  - opening connections to many hosts
- automated response:
  - router reconfigures to isolate suspicious host on its own subnet with access only to (e.g.) virus scanner download.
  - notifies administrators
- issue:
  - errors...



ALERTS

Term	Count	Action
SURICATA PROTO DETECT only one direction detected	720	🔍 ⊘
SURICATA ICMPv6 unknown type	336	🔍 ⊘
SURICATA PROTO DETECT first data in wrong direction	78	🔍 ⊘
SURICATA IPv6 HOPOPTS unknown option	56	🔍 ⊘
IPREP High Value SpywareCnC	12	🔍 ⊘
IPREP High Value Scanner	12	🔍 ⊘
SURICATA TLS invalid handshake message	8	🔍 ⊘
SURICATA STREAM excessive retransmissions	6	🔍 ⊘
SURICATA TLS invalid record/traffic	4	🔍 ⊘
ET POLICY curl User-Agent Outbound	2	🔍 ⊘
Other values	2	

TERMS

Term	Count	Action
Generic Protocol Command Decode	18	🔍 ⊘
Web Application Attack	2	🔍 ⊘
Attempted Information Leak	2	🔍 ⊘
Other values	1214	

USER-AGENTS

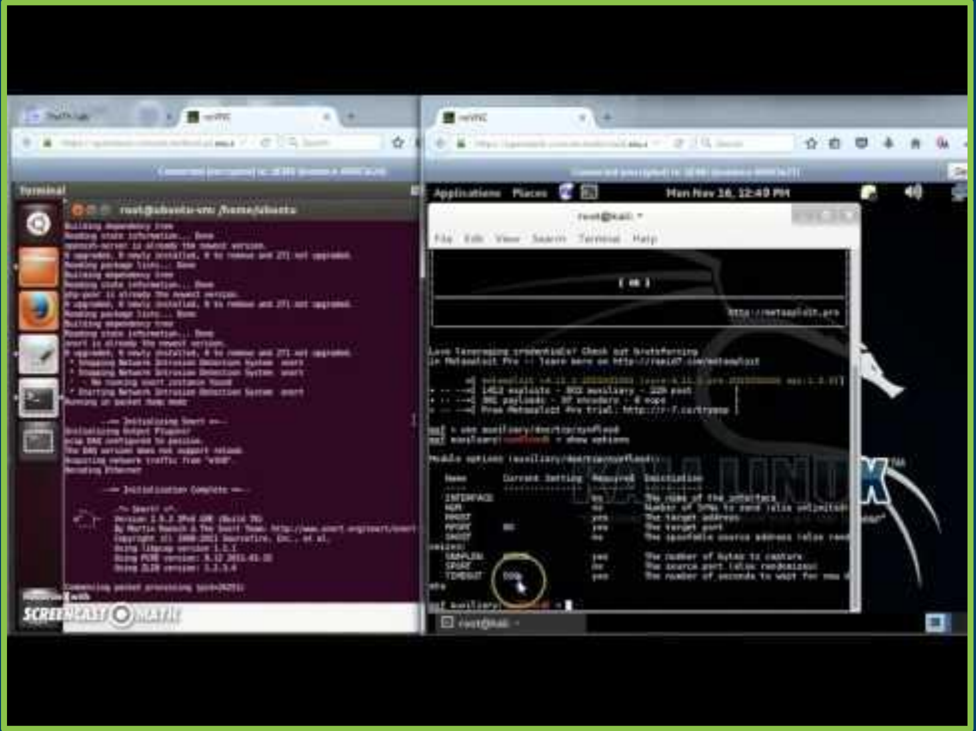
Term	Count	Action
Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:27.0) Gecko/20100101 Firefox/27.0	2486	🔍 ⊘
BW-C-2.0	1700	🔍 ⊘
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/32.0.1700.107 Chrome/32.0.1700.107	830	🔍 ⊘

TLS SUBJECT

Term	Count	Action
C=US, ST=California, L=San Francisco, O=Twitter, Inc., CN=*.twimg.com	200	🔍 ⊘
C=US, ST=California, L=Palo Alto, O=Facebook, Inc., CN=*.channel.facebook.com	144	🔍 ⊘
C=US, ST=California, L=San Francisco, O=Twitter, Inc., CN=si0.twimg.com	84	🔍 ⊘

Screenshot from “Suricata” (Network Intrusion Detection and Prevention tool)

Example: Intrusion Detection, Snort



```
Building dependency tree
Reading state information... Done
openssh-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 271 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
php-pear is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 271 not upgraded.
Reading package lists... Done
Building dependency tree
Reading state information... Done
snort is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 271 not upgraded.
 * Stopping Network Intrusion Detection System snort
 * Stopping Network Intrusion Detection System snort
 * - No running snort instance found
 * Starting Network Intrusion Detection System snort
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "eth0".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.2 IPv6 GRE (Build 78)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort
           Copyright (C) 1998-2011 Sourcefire, Inc., et al.
           Using libpcap version 1.1.1
           Using PCRE version: 8.12 2011-01-15
           Using ZLIB version: 1.2.3.4

Commencing packet processing (pid=[illegible])
```

```
                                   [ [illegible] ]

                                                          http://metasploit.pro

Love leveraging credentials? Check out bruteforcing
in Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.[illegible] ]
+ -- --=[ 1452 exploits - 802 auxiliary - 229 post ]
+ -- --=[ [illegible] payloads - 37 encoders - 8 nops ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else rand
omizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new d
ata

msf auxiliary(synflood) >
```

Network-Based Intrusion Detection System

typically separate machine

- **stealth mode:**
  - one NIC faces the network being monitored,
    no packets ever sent out on it,
    no packets can be routed specifically to it
  - another NIC faces a separate network
    through which alarms are sent
- **honeypot:**
  - dedicated machines(s) or networks
  - purpose is to look attractive to attacker
  - but actually just a trap: monitored to detect and surveil attacker

Errors

- **false positive**
  raise an alarm for a non-attack
  - makes administrators less confident in warnings
  - perhaps leading to actual attacks being dismissed
- **false negative**
  not raise an alarm for an attack
  - the attackers get in undetected!
- tradeoff between the two needs to be tunable;
  difficult to achieve the right classification statistics

(problem if both possible at the same time. cf. type soundness)

Intrusion Detection - Intrusion Detection System

Identification Methodologies

[Denning 1987]

- signature based:
  - recognize known attacks
- specification based:
  - recognize bad behavior
- anomaly based:
  - recognize abnormal behavior

Signature Based Detection

a.k.a. {misuse, rule-based} detection

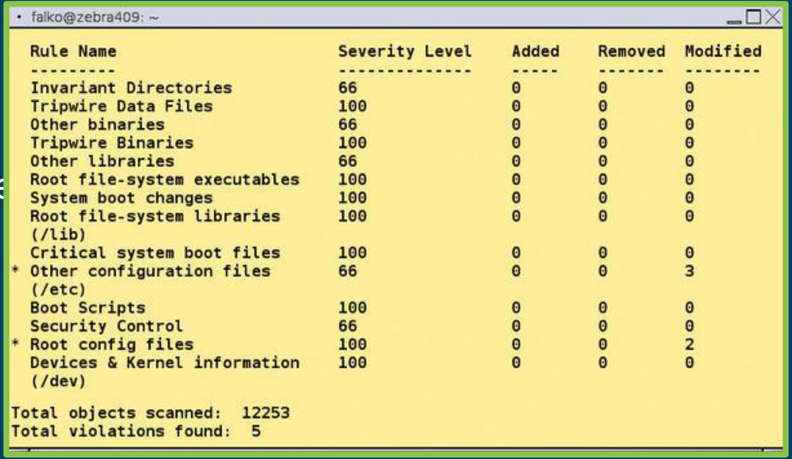
- characterize known attacks w/ signatures
- behavior matches signature ⇒ declare an intrusion
- issues:
  - works only for known attacks
  - signature needs to be robust w.r.t. small changes in attack

Intrusion Detection - Signature-Based Detection

Example: Tripwire

open source tool and commercial product

- **policy**:
  - certain files shouldn't change
- **state snapshot**:
  - analyzes filesystem, stores database of file hashe
- **automated response**:
  - runs (e.g. daily) and reports change of hash
- **issues**:
  - where to store database, how to protect its integrity, how to protect tripwire itself?



```
  Rule Name                        Severity Level    Added    Removed  Modified
  ---------                        --------------    -----    -------  --------
  Invariant Directories            66                0        0        0
  Tripwire Data Files              100               0        0        0
  Other binaries                   66                0        0        0
  Tripwire Binaries                100               0        0        0
  Other libraries                  66                0        0        0
  Root file-system executables     100               0        0        0
  System boot changes              100               0        0        0
  Root file-system libraries       100               0        0        0
  (/lib)
  Critical system boot files       100               0        0        0
* Other configuration files        66                0        0        3
  (/etc)
  Boot Scripts                     100               0        0        0
  Security Control                 66                0        0        0
* Root config files                100               0        0        2
  Devices & Kernel information     100               0        0        0
  (/dev)

Total objects scanned:  12253
Total violations found:  5
```

Intrusion Detection - Signature-Based Detection

Example: Network Flight Recorder

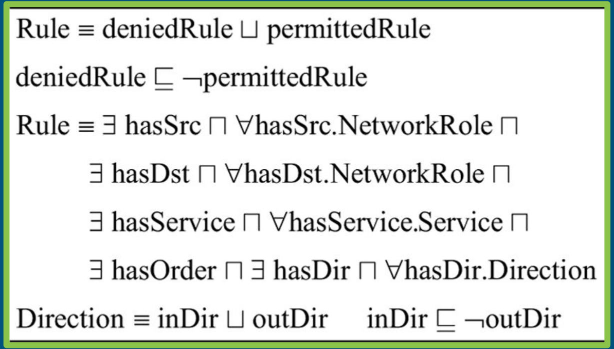
Network Flight Recorder

- three components:
  - packet sucker captures network traffic
  - decision engine uses custom-written filters in DSL to extract information from packets
  - backend writes information to disk; packets are discarded
- queries performed over stored information
  while rest of system continues to process packets
- similar ideas used in **Zeek** (aka. **Bro**)
  [Paxson 1999], available still as open source IDS

Machine	Count
I386	40
unknown-475	1

section_names.keyword: Descending	Count
.text	35
.data	34
.rsrc	32
.rdata	22

Specification Based Detection



$$\text{Rule} \equiv \text{deniedRule} \sqcup \text{permittedRule}$$

$$\text{deniedRule} \sqsubseteq \neg\text{permittedRule}$$

$$\begin{aligned}\text{Rule} \equiv\ & \exists\ \text{hasSrc} \sqcap \forall\text{hasSrc.NetworkRole} \sqcap \\ & \exists\ \text{hasDst} \sqcap \forall\text{hasDst.NetworkRole} \sqcap \\ & \exists\ \text{hasService} \sqcap \forall\text{hasService.Service} \sqcap \\ & \exists\ \text{hasOrder} \sqcap \exists\ \text{hasDir} \sqcap \forall\text{hasDir.Direction}\end{aligned}$$

$$\text{Direction} \equiv \text{inDir} \sqcup \text{outDir} \qquad \text{inDir} \sqsubseteq \neg\text{outDir}$$

- characterize good behavior of program w/ a specification
- if behavior ever departs from specification, declare an intrusion
- issues:
    - effort to create specifications
    - any program is a potential vulnerability if executed by a privileged user

Example: Distributed Monitor

[Ko et al. 1997]

- monitors Unix audit logs
- analyst writes grammar in DSL to describe good behavior
- parser checks conformance of logs with grammar
- distributed because it combines information from multiple hosts

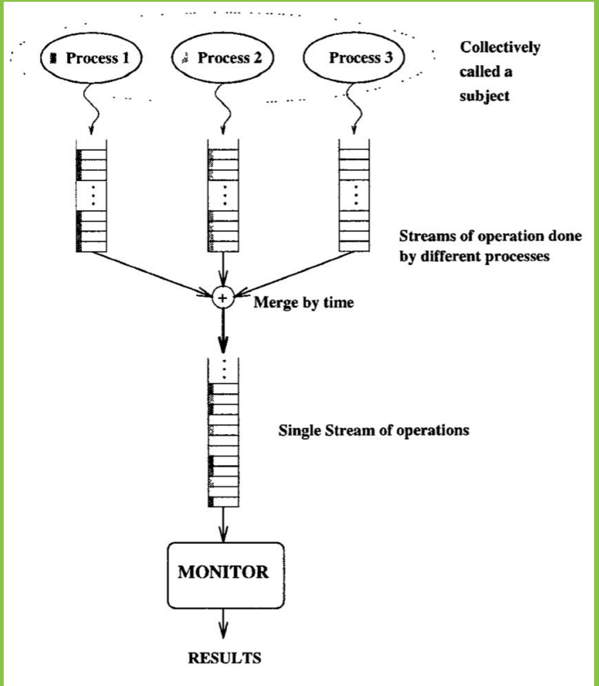
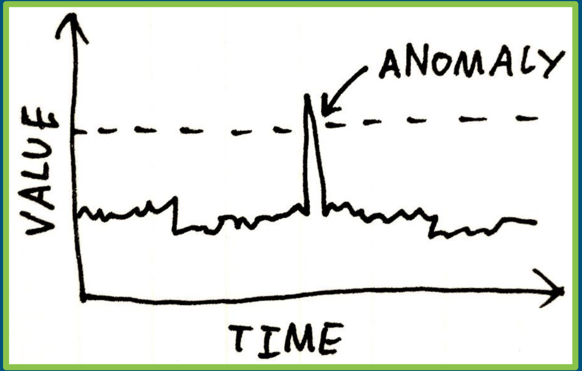

Process 1
Process 2
Process 3
Collectively called a subject
Streams of operation done by different processes
Merge by time
Single Stream of operations
MONITOR
RESULTS

Figure 1. The Monitoring Model.

Anomaly Based Detection



- characterize normal behavior of system
- if behavior ever departs far enough from normal, declare an intrusion
- issues:
  - feature identification
  - obtaining data on what is normal →

Example: Haystack (US Air Force)

[Smaha 1988] (influential; one of the earliest IDS papers)

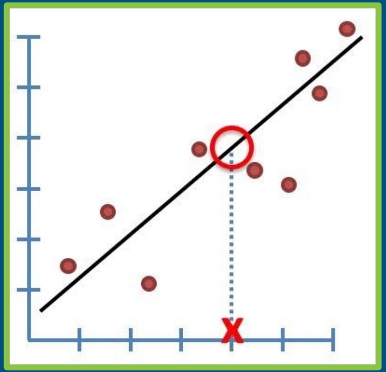
- monitors value of some statistic of interest over a sliding time window: a_i , a_{i+1}, ..., a_j ← (buffer j-i msg)
- determine lower and upper bounds t_L and t_U such that 90% of values lie between t_L and t_U
- next value is outside t_L and t_U ⇒ <u>anomaly</u>; raise alarm

ACSAC
WWW.ACSAC.ORG
Annual Computer Security Applications Conference
Test of Time Paper Award
Giovanni Vigna and Richard Kemmerer
"NetSTAT: A Network-based Intrusion Detection Approach"
Presented at ACSAC 35
San Juan, PR December 2019
ACSAC
WWW.ACSAC.ORG
Annual Computer Security Applications Conference
Test of Time Paper Award
Stephen Smaha
"Haystack: An Intrusion Detection System"
Presented at ACSAC 35
San Juan, PR December 2019
ACSAC
WWW.ACSAC.ORG
Annual Computer Security Applications Conference
Test of Time Paper Award
Michael Reed, Paul Syverson, and David Goldschlag
"Proxies for Anonymous Routing"
Presented at ACSAC 35
San Juan, PR December 2019

adaptive

- as time passes, window moves, so detector adjusts itself.

Statistical Models



ML great for *classification*...

- **threshold models**
  - min and max
- **moment models**
  - mean & standard deviation
- **markov models**
  - probability of next event based on current state

BUT, ML **not** great for *outlier detection*. Adversarial ML poorly understood.

Intrusion Response?

intrusion handling: [Northcutt 1998]

1. Preparation
2. Identification
3. Containment
4. Eradication
5. Recovery
6. Follow up

automated response: monitor, protect, alert

counterattack?

- **legal route:** file criminal complaint
- tech route: damage attacker
  - might harm innocents
  - might expose you legally

Detect Attacks: Past

Damage is done.

How will we know?

How will we handle it?

- Auditing
- Anti-malware

Auditing

What to Log?

example: US State Dept. pilot program (1980s)

- requirements
  - log every transaction related to protected electronic documents
  - system administrator reviews log daily to search for malicious behavior
- experiment
  - test system for 5 users, 10 minutes
- result
  - audit log = stack of paper over 1ft high
  - real system would have been 1000s of users working 24/7
- lessons learned
  - logging and review of everything by a human is impractical
  - need: reduce information logged
  - need: automated review

Auditing - What to Log

States vs. Events

States

data; what system *is*

- backup
- more?

what state to log?

pros: survive power failures, crashes, attacks.
cons: what state? memory, disk, network, …
what about distributed systems? (hard)

Events

our focus

actions, how system *came to be*

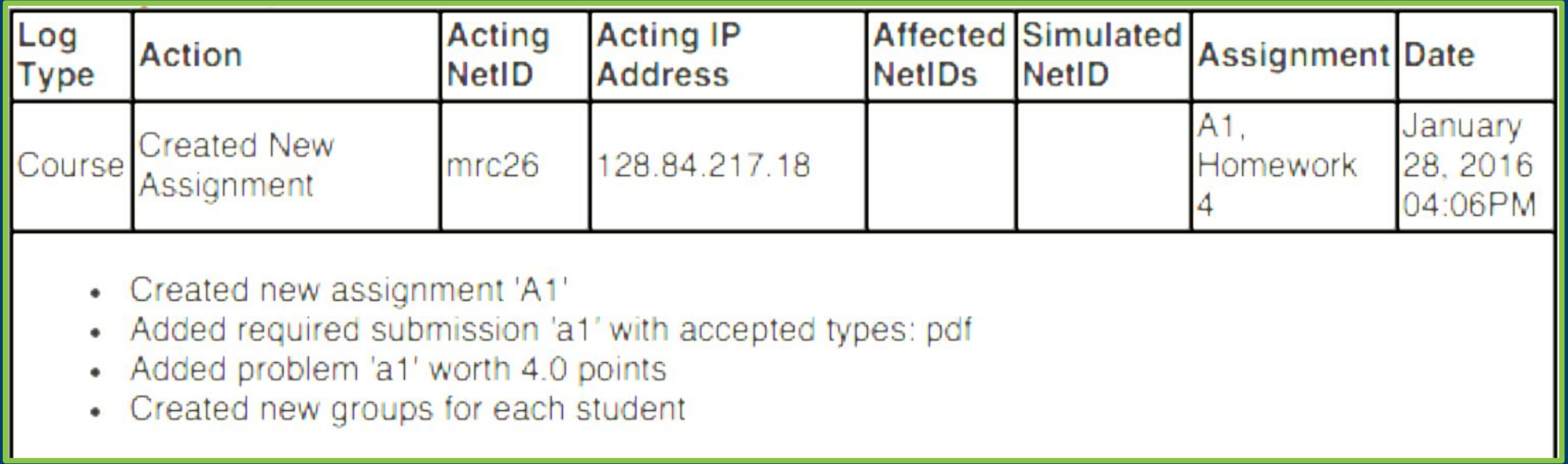
- login
- access to protected resource,
- elevation and attenuation of privileges,
- ...

which events to log?

- event relevant for security
- what check was made, outcome, information that lead to that decision.

In-Class: Course Management System

what <u>kind of events</u> to log for a course management system (*mutations*)?
what <u>details</u> would you put into the log entry?



Log Type	Action	Acting NetID	Acting IP Address	Affected NetIDs	Simulated NetID	Assignment	Date
Course	Created New Assignment	mrc26	128.84.217.18			A1, Homework 4	January 28, 2016 04:06PM

- Created new assignment 'A1'
- Added required submission 'a1' with accepted types: pdf
- Added problem 'a1' worth 4.0 points
- Created new groups for each student

Auditing - What to Log

In-Class: Course Management System

Course

- add students,
- change group timeslot,
- computed assignment stats,
- computed total score,
- created / edited removed / restored assignment,
- created / removed / restored announcement,
- created / removed timeslot,
- dropped students,
- edited course properties,
- edited staff preferences,
- edited student preferences,
- sent course email
- uploaded class list

Content

- added / edited content data,
- create / edited / reorder / remove content,
- add students,
- change group timeslot,

Group

- sent / canceled group invite
- joined / left group
- created / disbanded group
- granted / removed extension
- requested regrade,

Grade

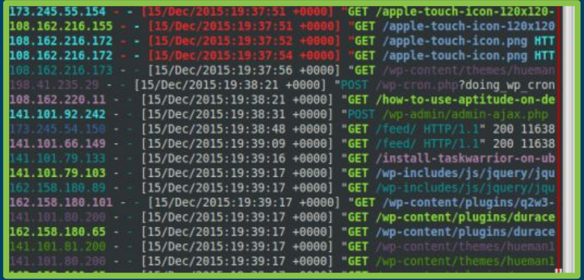
- assigned grader
- edited grades
- edited comments
- uploaded grade files

Auditing - What to Log

In-Class: Course Management System

details logged:

- event type
- acting NetID
- acting IP address
- affected NetIDs
- simulated NetID
- assignment, if any
- event details (no sanitization of grades)



```
173.245.55.154 - - [15/Dec/2015:19:37:51 +0000] "GET /apple-touch-icon-120x120-
108.162.216.155 - - [15/Dec/2015:19:37:51 +0000] "GET /apple-touch-icon-120x120
108.162.216.172 - - [15/Dec/2015:19:37:52 +0000] "GET /apple-touch-icon.png HTT
108.162.216.172 - - [15/Dec/2015:19:37:54 +0000] "GET /apple-touch-icon.png HTT
108.162.216.173 - - [15/Dec/2015:19:37:56 +0000] "GET /wp-content/themes/hueman
198.41.235.29 - - [15/Dec/2015:19:38:21 +0000] "POST /wp-cron.php?doing_wp_cron
108.162.220.11 - - [15/Dec/2015:19:38:21 +0000] "GET /how-to-use-aptitude-on-de
141.101.92.242 - - [15/Dec/2015:19:38:31 +0000] "POST /wp-admin/admin-ajax.php
173.245.54.150 - - [15/Dec/2015:19:38:48 +0000] "GET /feed/ HTTP/1.1" 200 11638
141.101.66.149 - - [15/Dec/2015:19:39:09 +0000] "GET /feed/ HTTP/1.1" 200 11638
141.101.79.133 - - [15/Dec/2015:19:39:16 +0000] "GET /install-taskwarrior-on-ub
141.101.79.103 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-includes/js/jquery/jqu
162.158.180.89 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-includes/js/jquery/jqu
162.158.180.101 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-content/plugins/q2w3-
141.101.80.200 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-content/plugins/durace
162.158.180.65 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-content/plugins/durace
141.101.81.200 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-content/themes/hueman1
141.101.80.200 - - [15/Dec/2015:19:39:17 +0000] "GET /wp-content/themes/hueman1
```

How to Log

Say what you mean

log entry should **say what it means**.

- interpretation of log entry should depend only on content of log entry
- ⇒ reviewer can recover meaning w/o needing to assume / supply context
- good practice: write down straightforward English sentence describing the meaning of each log entry

Auditing - How to Log

Standard Log File Format

keeping log files in standard format enables...

- reuse of tools for log analysis
- correlation across logs from multiple applications

standard formats:

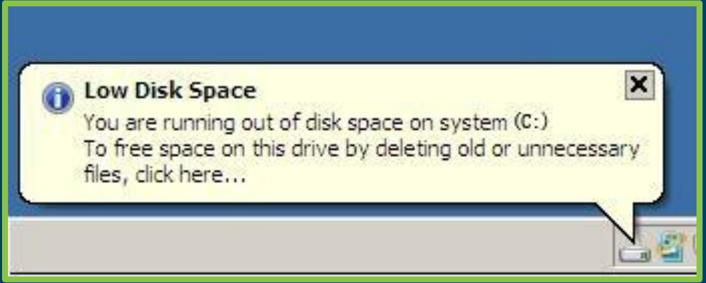
- Common Log Format (NCSA; used by web servers)
- **syslog** (used by Unix)
  - originated with sendmail
  - became de facto standard
  - then standardized by IETF: RFC 5424
  - examples: take a look in your local /var/log directory

Auditing - How to Log

Log Size Too Large?

what happens if log size grows too large?

- stop logging
- overwrite previous entries
- halt system



Low Disk Space

You are running out of disk space on system (C:)
To free space on this drive by deleting old or unnecessary files, click here...

all used in practise, depending on scenario.

(none of these options are great. but you have to do something)

Auditing

Manual

enable admins to explore logs and look for {states,events}.

issues:

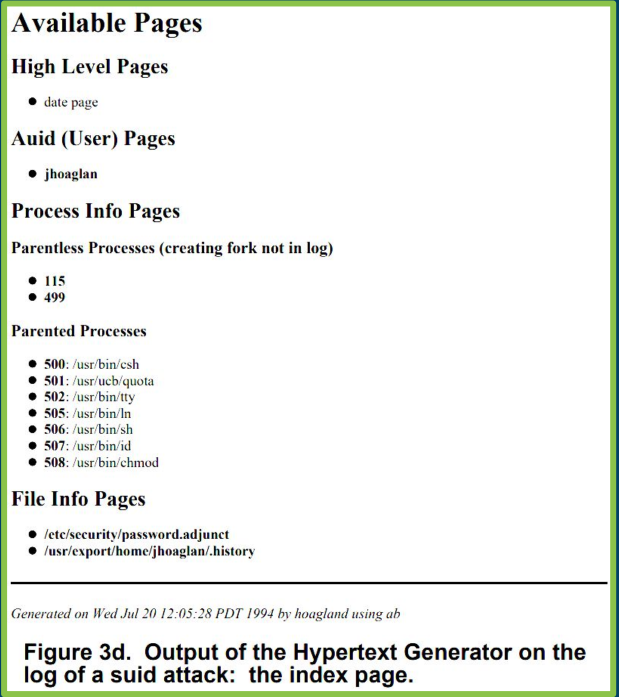
- designers might not have recorded the right {states,events}
- visualization, query, expressivity (HCI/DB issues)
- correlation amongst multiple logs

Auditing - Manual

Visualization

Interface

- text
  - example: syslog (previous slide)
- hypertext
- DBMS
  - example: queries in the course management system
- graph
  - nodes might be entities (processes, files), edges might be associations (forking, times)



Available Pages

High Level Pages

- date page

Auid (User) Pages

- jhoaglan

Process Info Pages

Parentless Processes (creating fork not in log)

- 115
- 499

Parented Processes

- **500**: /usr/bin/csh
- **501**: /usr/ucb/quota
- **502**: /usr/bin/tty
- **505**: /usr/bin/ln
- **506**: /usr/bin/sh
- **507**: /usr/bin/id
- **508**: /usr/bin/chmod

File Info Pages

- /etc/security/password.adjunct
- /usr/export/home/jhoaglan/.history

Generated on Wed Jul 20 12:05:28 PDT 1994 by hoagland using ab

Figure 3d. Output of the Hypertext Generator on the log of a suid attack: the index page.

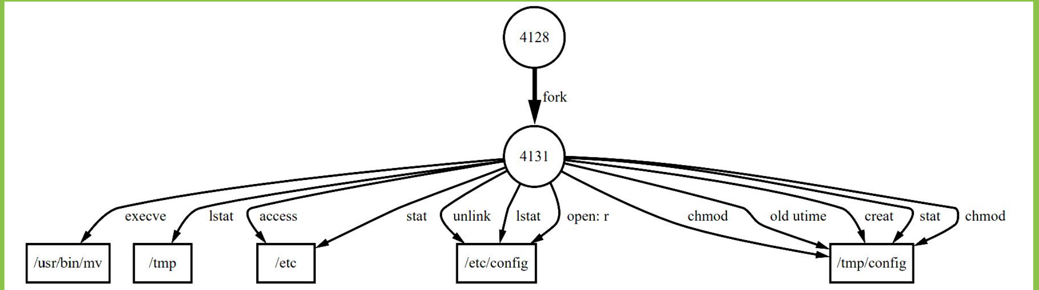


Figure 4b. FAB presentation of "Investigating the Disappearance of a File" situation with focus on process 4131. The process that removed the file, 4131, is now investigated further with other system calls and its parent process shown. It is seen that process is running the "mv" program, which is show to read and remove "/etc/config",

Auditing - Manual

Visualization

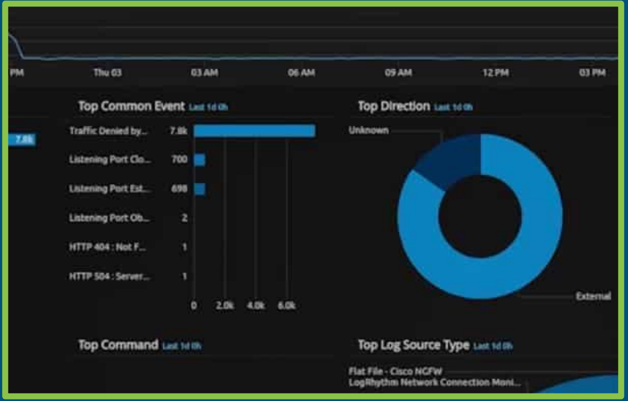
Interface

- text
  - example: syslog (previous slide)
- hypertext
- DBMS
  - example: queries in the course management system
- graph
  - nodes might be entities (processes, files), edges might be associations (forking, times)

Technique

- temporal
  - animate what happened and when (e.g. time-ordered sequence of graphs)
- slice
  - minimal set of log events that affected an object

Automatic



example: LogRhythm

detect

- suspicious behavior
- violations of explicit policy

built how

- custom-built systems
- classic AI techniques like training neural nets, expert systems, etc.
- machine learning

response: monitor, report, take action

e.g. close account / connection

Malware removal

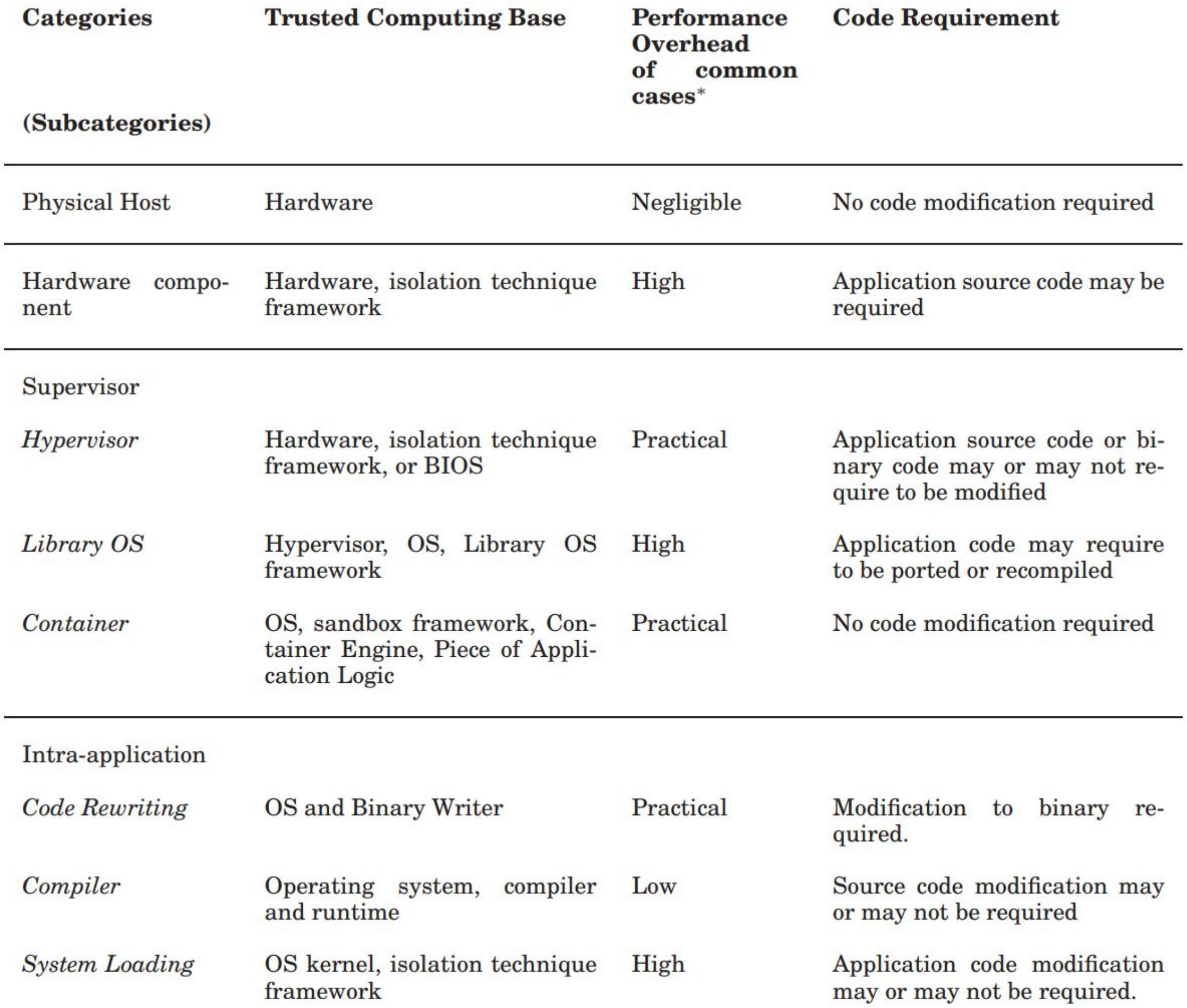
antivirus

Administration

What can Sys-Admins do, to secure systems, w/o writing them themselves?

- Hardening
- Firewalls
- Isolation

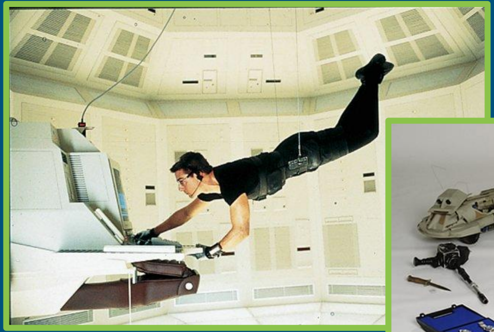
Isolation Categories



Categories (Subcategories)	Trusted Computing Base	Performance Overhead of common cases*	Code Requirement
Physical Host	Hardware	Negligible	No code modification required
Hardware component	Hardware, isolation technique framework	High	Application source code may be required
Supervisor			
Hypervisor	Hardware, isolation technique framework, or BIOS	Practical	Application source code or binary code may or may not require to be modified
Library OS	Hypervisor, OS, Library OS framework	High	Application code may require to be ported or recompiled
Container	OS, sandbox framework, Container Engine, Piece of Application Logic	Practical	No code modification required
Intra-application			
Code Rewriting	OS and Binary Writer	Practical	Modification to binary required.
Compiler	Operating system, compiler and runtime	Low	Source code modification may or may not be required
System Loading	OS kernel, isolation technique framework	High	Application code modification may or may not be required.

isolation done where:
outside the computer

Physical



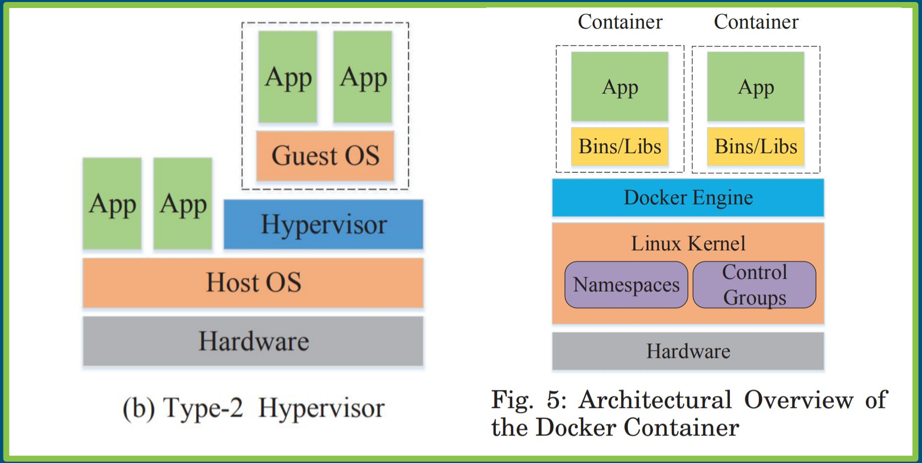
air-gap: system physically isolated from network.

how to breach (w/ enough **resources**)?
Hollywood-level creative.

Isolation

isolation done where:
outside the program

Supervisor



(b) Type-2 Hypervisor

Fig. 5: Architectural Overview of the Docker Container

hypervisor (VMM): hosts a computer, in software, on which SW runs.

- Type-1 (Hyper-V, Xen)
- Type-2 (VirtualBox, VMware)

library OS: OS as user-mode library. “Multiple OS” on a shared system.

- Graphene

containerization: restricted execution environment.

- `chroot, docker, lxc`

Isolation

for completeness; sys-admins don’t really ever do this

isolation done where:
inside the program

Intra-Application

code-rewriting: rewrite code to introduce isolation into it.

monitors, binary instrumentation

compiler: program analysis rejects program that do not have isolation.

Java memory safety, CompCert
IFC languages

system-loading: force system to use customized libs that do access control

`boxify`

Summary

Summary

Arms Race

now you know some of what **Security Analysts**, **Forensic Analysts**, and **System Administrators** do.

- detect vulnerabilities
  tools: *scan*, look up *CVE*
- audit for attacks
  how/who/when/where
  - manual: visualization
  - automatic: report/act
- security w/o building security in

adblock,
fuckadblock
fuckfuckadblock,
...

important: limits; e.g. pattern-based approaches can be circumvented