The Linux Audit System

WAJIH
04/30/2018
$whoami

- Third year Ph.D. student in CS Dept.
- Working with Prof. Adam Bates
- Research Interests:
  - System Security
  - Data provenance
Recent Cyber Attacks

- **Equifax**
  - 145 million Americans’ sensitive data (e.g. SSN) was stolen

- **WannaCry**
  - A ransomware attack that spans over 150 countries
  - Hackers demanded money to unlock files

- **A Yahoo bombshell**
  - Yahoo's 3 billion accounts was hacked in 2013 – found out in 2016
Recent Cyber Attacks

- **Equifax**
  - 145 million American's sensitive data (e.g., SSN) was stolen

- **WannaCry**
  - Aransomeware that spans over 150 countries
  - Hackers demanded money to unlock files

- **A Yahoo bombshell**
  - Yahoo's 3 billion accounts was hacked in 2013 – found out in 2016

---

**Advanced Persistent Threat (APT)**

Targeted: Targets specific organizations to exfiltrate information or disrupt the systems.
5 Stages of APTs

1. **Reconnaissance**
   - Understand about the target using social media or company’s website

2. **Incursion**
   - Enters into victim’s system using different attack vectors (e.g. social engineering)

3. **Discovery**
   - The attackers stay low and operate patiently in order to avoid detection

4. **Capture**
   - Hackers access unprotected systems and capture data over an extended period of time

5. **Exfiltration**
   - Finally, captured information is sent back to the attack team’s home base for analysis
5 Stages of APTs

1. **Reconnaissance**
   - Understand about the target using social media or company's website

2. **Incursion**
   - Enters into victim's system using different attack vectors (e.g., social engineering)

3. **Discovery**
   - Hackers stay low and operate patiently to avoid detection

4. **Capture**
   - Hackers access unprotected systems and capture data over an extended period of time

5. **Exfiltration**
   - Finally, captured information is sent back to the attack team’s home base for analysis

*Due to complexity of APTs*

Attack investigation such as finding root cause is challenging
Audit Logging Or Data Provenance

- Attack investigation and reconstruction technique
- Captures data life cycle:
  - Modifications
  - Deletions
  - Creations
- Detects causal dependencies between different events
Example Audit Log

- chromium.exe reads from ip 10.0.0.2
- chromium.exe reads from ip 165.10.0.1
- chromium.exe reads from ip 91.0.0.2
- chromium.exe downloads a.ppt
- chromium.exe downloads b.doc
- chromium.exe downloads malware.exe
- malware.exe reads /etc/passwd
- malware.exe sends /etc/passwd to ip X.X.X.X
Represented as causal graph

- Vertices represents system entities (e.g. chrome process, a.ppt)
- Edges represents causal relationships (e.g. created, read, open)
Linux Audit System

- Linux Audit System collects audit logs
- Available on vanilla Linux kernels > version 2.6
- It collects information regarding:
  - Kernel event (System calls)
  - User events (Audit-enable programs)
  - It does not, however, provide additional security itself—it does not protect your system from code malfunctions
Linux Audit Use cases

- **Watching file access:**
  - Audit can track whether a file or a directory has been accessed, modified, executed

- **Monitoring system calls:**
  - Generate a log entry every time a particular system call is used

- **Recording commands run by a user:**

- **Monitoring network access:**
  - The **iptables** and **ebtables** utilities can be configured to trigger Audit events
How Linux Audit Works?

- Audit kernel module intercepts the system calls and records the relevant events.
- The auditd daemon writes the audit reports to disk.
- Various command line utilities take care of displaying, querying, and archiving the audit trail.
How Linux Audit Works?
Components of Linux Audit

- **auditctl** — utility for managing the auditd daemon; returns information on the audit subsystem’s current status and can be used to add and delete rules
- **ausearch** — utility for searching for events in log files
- **aureport** — utility for generating reports on the audit system
Components of Linux Audit
Creating rules

- auditctl is command line utility to:
  - Control behaviour of audit daemon (auditd)
  - Add and remove audit rules

- There are two main types of rules:
  - File system audit rules
  - System call audit rules
File System Rules

- File System rules are sometimes called watches.
- These rules are used to audit access to particular files or directories that you may be interested in.
- The syntax of these rules generally follow this format:
  \[-w\] path-to-file \[-p\] permissions \[-k\] keyname
- where the permission are any one of the following:
  - r - read of the file
  - w - write to the file
  - x - execute the file
  - a - change in the file's attribute
System call rules

- The system call rules are loaded into a matching engine that intercepts each syscall that all programs on the system makes.
- Very important to only use syscall rules when you have to since these affect performance
- Syscall rules take the general form of:
  -a action,list -S syscall -F field=value -k keyname

- To see files opened by a specific user:
  -a exit,always -S open -F auid=l337
- To see unsuccessful open calls:
  -a exit,always -S open -F success=0
Example

- Track a file by inode number
  ```
  # auditctl -a exit,always -S open -F inode=`ls -i /etc/auditd.conf | gawk '{print $1}'`
  # auditctl -l
  AUDIT_LIST: exit,always inode=1637178 (0x18b3a) syscall=open
  ```

- When someone opens the files you receive following log message

  type=PATH msg=audit(1251123553.303:206): item=0 name="/etc/audit/audit.rules"
  inode=77546 dev=fd:01 mode=0100640 ouid=0 ogid=0 rdev=00:00
  obj=system_u:object_r:auditd_etc_t:s0
Analyzing logs -- ausearch

- Ausearch is a command-line utility to query your audit logs
- ausearch -f
- ausearch -ui
Analyzing logs - aureport

```
$ sudo aureport -s

Syscall Report
=============
# date time syscall pid comm auid event
===============
1. 08/03/2015 15:45:03 313 10285 modprobe -1 52501
2. 08/03/2015 15:45:03 313 10290 modprobe -1 52502
3. 08/03/2015 15:45:03 54 10296 iptables -1 52503
4. 08/03/2015 15:45:03 54 10302 iptables -1 52504
5. 08/03/2015 15:45:03 54 10305 iptables -1 52505
6. 08/03/2015 15:45:03 54 10313 iptables -1 52506
7. 08/03/2015 15:45:03 54 10325 iptables -1 52507
8. 08/03/2015 15:45:03 54 10329 iptables -1 52508
9. 08/03/2015 15:45:03 54 10343 iptables -1 52509
10. 08/03/2015 15:45:03 54 10345 iptables -1 52510
11. 08/03/2015 15:45:03 54 10349 iptables -1 52511
```
Audit Data Visualization

- Various tools to generate causal graphs from audit logs.
- I use SPADE tool
- SPADE (https://github.com/ashish-gehani/SPADE)
  - Parses audit log in realtime
  - Generates causal graphs which can be queried to find the root cause of attack
Audit Data Visualization

- ADD here about SPADE tools
Resources

- The Audit Manual Pages:
  - There are several man pages installed along with the audit tools that provide valuable and very detailed information

  - The home page of the Linux audit project.