Attack Tolerant Software (Systems)

Mladen Vouk
Professor
Engineering Attack-Tolerant Software (ATS)

• **Applied science** of measuring, engineering, managing and predicting attack resilience of software-based systems to maximize customer satisfaction.

• **Project Goal**: Study scientific principles behind ATS by extending science behind software fault-tolerance engineering (FTE).
Uncertainty

• **Epistemic** - incorrect *knowledge or process* (e.g., erroneous models, approximations, systematic errors, etc.) – root cause

• **Aleatoric** – due innate *randomness* (accidents, zero-day exploits, unknown, typos, …)
Causes

• **Software fault causes:**
  - communications (mix),
  - knowledge (epistemic),
  - incomplete analysis (mix), and
  - transcription (aleatoric).

• **Note:** Most of CWE top 25 errors are epistemic.
Engineering Software

• Fault Avoidance and Removal Process
  – Removal: Fault identification and correction
  – Proof-of-program-correctness is currently impractical for real systems.
  – Exhaustive testing is currently impractical for real systems.
  – Faults (including vulnerabilities) due to both epistemic and aleatoric events are inevitable.
Fault-Tolerance

• Ability to recognize anomalies and effect recovery **without** allowing an exploit
  – Forward recovery
  – Backward recovery
  – “Graceful” (safe) failure

• Hinges on anomaly recognition unless continuous forward error correction or masking (regardless of state) is used.
Human Component

• On good days human error rate is 0.00001, under stress it can be as high as 1 in 10.
• It is important to reduce, as much as possible, human errors (how-to-use errors).
  • Phishing
  • Weak passwords
  • Incorrect configurations
  • Etc.
Anomaly Detection

- Acceptance testing (invariably domain specific, internal frame of reference)
- Consistency checking (external reference frame, situation awareness, usually not available at run-time, off-line tool)
- Voting (practical, expensive, correlation?)
- Watchdog (e.g., time-out, race) – practical (situation awareness, can be disruptive)
Acceptance Testing (Epistemic)

- Passwords (strong, weak)
- Firewalls (access rights, b/w lists, ...) – stat, dyn
- Input verification (e.g., dangerous characters) – SQLi, CLIi, XSS, ...
- Intrusion detection devices (patterns, ...)
- Error propagation, Algorithmic, e.g.,
  \[\text{for } (0 \leq x \leq y) \ (\text{Abs}((\text{Sqrt}(x)\times\text{Sqrt}(x))-x)<E)\]
- Checksums
- etc.
Redundancy and Voting

- Redundancy – identical backups, algorithmic/programmer diversity, obfuscation diversity, ...

- Adjudication by Voting (common cause faults, failure independence, ..)
  - Majority voting
  - Consensus voting
  - Voting in small spaces
  - Other
Recovery Block

- Rejuvenation, all acceptance-based approaches, …, RB based designs, etc.
N-Version Programming

- Cloud friendly, Back-to-back testing bonus, functionally equivalent (FE) alternatives.

```
Input

FE1  FE2  FEn

Success

Voter

Exception

Common Cause Failure Correlation?
```
Modeling Attacks

N=20

Ad Hoc (Aleatoric)

Epistemic, HG Attack
Sampling w/o replacement

Operational-like (Mix)
Sampling with replacement

Vulnerabilities

Attack Cases

Science of Security
Lablet

Resilient
Architectures

Computer Science
NC State University
HyperGeometric Attack

• Assumption: there is a limited amount of time and resources an attacker is ready to spend (i.e., there will be a limited number of attack/test cases thrown at the target).

• Attack sequences will target specific real or imagined vulnerabilities by (importance) sampling the attack space without replacement (i.e., not repeating attack or test cases).
An Actual Short Epistemic Web Attack Burst

Normal requests …
Attack starts (nominally from a Netherlands address)
1.226.83.165 - - [26/Jan/2013:08:39:34 -0500] "GET /w00tw00t.at.blackhats.romanian.anti-sec:) HTTP/1.1" 404 247
1.226.83.165 - - [26/Jan/2013:08:39:36 -0500] "GET /admin/scripts/setup.php HTTP/1.1" 404 229

Attack ends and normal requests continue…
Actual Short Epistemic Web Attack Burst (robot probe)

Normal requests …

Attack starts (nominally from a Netherlands address)

1.226.83.165 - - [26/Jan/2013:08:39:34 -0500] "GET /w00tw00t.at.blackhats.romanian.anti-sec:) HTTP/1.1" 404 247
1.226.83.165 - - [26/Jan/2013:08:39:36 -0500] "GET /admin/scripts/setup.php HTTP/1.1" 404 229

Attack ends and normal requests continue…

I know this is an attack because or
I will change the name to goahead7.php

I have engineered away PHP functionality.

Security by Obscurity. How many holes I have not plugged?
Samples of Actual Aleatoric SSH “Attacks”

Dec 11 21:22:16 renoir sshd[13276]: login to account j jj not allowed or account non-existent.
Dec 11 22:11:43 renoir sshd[13108]: login to account v ok not allowed or account non-existent.
Dec 13 14:56:58 renoir sshd[13042]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:51483', requested service 'ssh-connection')
Dec 13 14:58:20 renoir sshd[13044]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:51488', requested service 'ssh-connection')
Dec 14 08:08:16 renoir sshd[13848]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:54241', requested service 'ssh-connection')
Samples of Actual Aleatoric SSH “Attacks”

Dec 11 21:22:16 renoir sshd[13276]: login to account jjj not allowed or account non-existent.
Dec 11 22:11:43 renoir sshd[13108]: login to account vok not allowed or account non-existent.
Dec 13 14:56:58 renoir sshd[13042]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:51483', requested service 'ssh-connection')
Dec 13 14:58:20 renoir sshd[13044]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:51488', requested service 'ssh-connection')
Dec 14 08:08:16 renoir sshd[13848]: Remote host disconnected: Authentication method disabled. (user 'streck', client address '152.14.52.222:54241', requested service 'ssh-connection')

I know this is just an accident because?

or

This is real

Known probability of guessing the password is P (science).
Sampling

Metric: M
Space size: K
Coverage: K-u(i)
Per case: h(i)

Vulnerability
Model

Vulnerabilities that remain undiscovered after H/G attack is over

\[ E_i = N - N \left( 1 - \prod_{j=1}^{i} \left( 1 - \frac{g_j \cdot h_j}{u_i} \right) \right) \]

• Attack case i covers \( h(i) \geq 1 \) new constructs, out of the total of \( K \), that under metric \( M \) might expose one or more vulnerabilities (case efficiency is \( g(i) \geq 1 \)). After case i, \( u(i) \) uncovered constructs remain. In an ideal situation one needs not more cases than the there are vulnerabilities (total of \( N \) vulnerabilities).

• If attacker’s \( M \) and tactics are known, one can estimate probability that attack will be successful and act on that.
Directions and Collaborations

• Assessment of H/G and other models in industrial setting
• Development of appropriate dynamic attack tolerance responses based on attack classification
• Recognition of stealth (noise level) attacks
• Data and interactions
• Other