Lecture 29
December 8, 2023

Anti-Forensics

- *Anti-forensics*: the attempt to compromise the availability or usefulness of evidence to forensics process
- Goals:
  - Interfere with forensic analysis tools gathering information, by hiding data or obscuring type, sequence of evidence
  - Hinder the validation of authenticity of digital image
  - Exploit weaknesses in forensic analysis tools
  - Attacking users of forensic analysis tools, for example by crashing analyst's system or increasing time needed to analyze data
  - Cast doubt on results of forensic analysis; will diminish its credibility in court, for example

Examples

- *timestomp*: enables user to change file access times
- *event_manager*: enables user to delete entries from log files
- JPEG image data compresses digital representation of image into multiple bands of transform coefficients, which generally follow a smooth distribution; altering image perturbs coefficients, so distribution different; anti-forensic tools add dithering to change coefficients back to approximate original one
- Forensic tool determines if Windows files are executable by looking at file extension (".exe") and first 2 bytes of file ("MZ"), so anti-forensics tools can just change the extension

Intrusion Detection

- Detect wide variety of intrusions
  - Previously known and unknown attacks
  - Suggests need to learn/adapt to new attacks or changes in behavior
- Detect intrusions in timely fashion
  - May need to be be real-time, especially when system responds to intrusion
    - Problem: analyzing commands may impact response time of system
  - May suffice to report intrusion occurred a few minutes or hours ago

Intrusion Detection Systems

- Present analysis in simple, easy-to-understand format
  - Ideally a binary indicator
  - Usually more complex, allowing analyst to examine suspected attack
  - User interface critical, especially when monitoring many systems
- Be accurate
  - Minimize false positives, false negatives
  - Minimize time spent verifying attacks, looking for them

Principles of Intrusion Detection

- Characteristics of systems not under attack
  - User, process actions conform to statistically predictable pattern
  - User, process actions do not include sequences of actions that subvert the security policy
  - Process actions correspond to a set of specifications describing what the processes are allowed to do
- Systems under attack do not meet at least one of these

Example

- Goal: insert a back door into a system
  - Intruder will modify system configuration file or program
- Requires privilege; attacker enters system as an unprivileged user and must acquire privilege
  - Nonprivileged user may not normally acquire privilege (*anomaly*)
  - Attacker may break in using sequence of commands that violate security policy (*misuse*)
  - Attacker may cause program to act in ways that violate program’s specification (*specification*)

Denning’s Model

- Hypothesis: exploiting vulnerabilities requires abnormal use of normal commands or instructions
  - Includes deviation from usual actions
  - Includes execution of actions leading to break-ins
  - Includes actions inconsistent with specifications of privileged programs

Models of Intrusion Detection

- Anomaly detection
  - What is usual, is known
  - What is unusual, is bad
- Misuse detection
  - What is bad, is known
  - What is not bad, is good
- Specification-based detection
  - What is good, is known
  - What is not good, is bad

Anomaly Detection

- Analyzes a set of characteristics of system, and compares their values with expected values; report when computed statistics do not match expected statistics
  - Threshold metrics
  - Statistical moments
  - Markov model

Types of Learning

- *Supervised learning methods*: begin with data that has already been classified, split it into “training data”, “test data”; use first to train classifier, second to see how good the classifier is
- *Unsupervised learning methods*: no pre-classified data, so learn by working on real data; implicit assumption that anomalous data is small part of data
- Measures used to evaluate methods based on:
  - TP: true positives (correctly identify anomalous data)
  - TN: true negatives (correctly identify non-anomalous data)
  - FP: false positives (identify non-anomalous data as anomalous)
  - FN: false negatives (identify anomalous data as non-anomalous)

Misuse Detection

- Determines whether a sequence of instructions being executed is known to violate the site security policy
  - Descriptions of known or potential exploits grouped into *rule sets*
  - IDS matches data against rule sets; on success, potential attack found
- Cannot detect attacks unknown to developers of rule sets
  - No rules to cover them

Specification Modeling

- Determines whether execution of sequence of instructions violates specification
- Only need to check programs that alter protection state of system
- System traces, or sequences of events t_1, … t_i, t_{i+1}, …, are basis of this
  - Event t_i occurs at time $C(t_i)$
  - Events in a system trace are totally ordered

Comparison and Contrast

- Misuse detection: if all policy rules known, easy to construct rulesets to detect violations
  - Usual case is that much of policy is unspecified, so rulesets describe attacks, and are not complete
- Anomaly detection: detects unusual events, but these are not necessarily security problems
- Specification-based vs. misuse: spec assumes if specifications followed, policy not violated; misuse assumes if policy as embodied in rulesets followed, policy not violated

Measuring Effectiveness

- *Accuracy*: percentage (or fraction) of events classified correctly
  - ((*TP* + *TN*) / (*TP* + *TN* + *FP* + *FN*)) * 100%
- *Detection rate*: percentage (or fraction) of reported attack events that are real attack events
  - (*TP* / (*TP* + *FN*)) * 100%
  - Also called the *true positive rate*
- *False alarm rate*: percentage (or fraction) of non-attack events reported as attack events
  - (*FP* / (*FP* + *TN*)) * 100%
  - Also called the *false positive rate*

Usefulness of Measurement

- Data at installation should be similar to that used to measure effectiveness
- Example: military, academic network traffic different
  - KDD-CUP-99 dataset derived from unclassified and classified network traffic on an Air Force Base
  - Network data captured at Florida Institute of Technology
- FIT data showed anomalies not in KDD-CUP-99
  - FIT data: TCP ACK field nonzero when ACK flag not set
  - KDD-CUP-99 data: HTTP requests all regular, all used GET, version 1.0; in FIT data, HTTP requests showed inconsistencies, some commands not GET, versions 1.0, 1.1
- Conclusion: using KDD-CUP-99 data would show some techniques performing better than they would on the FIT data

Computers and Elections

- This looks at the technology
  - Procedures, policies equally important, but require a different type of analysis (“process modeling”, used to model software development, can be applied here)
- Does using computers in an election process:
  - Introduce new ways for attackers to compromise the election, or prevent voters from voting?
  - Stop any of the previous ways for attackers to compromise the election, or provide new ways to enable voters to vote?

Some Terms for E-Voting Systems

- BMD: Ballot Marking Device
  - Marks a paper ballot
- DRE: Direct Recording Electronic
  - Stores votes (ballots) electronically
- DRE + VVPAT: DRE + Voter Verified Paper Audit Trail
  - A DRE that also prints a paper record of the votes (ballots) cast on it
- PCOS: Precinct Count Optical Scanners
  - Used to count paper ballots at the precinct (polling station); these are stored electronically and the memory cards used to transfer results to central vote tabulator

Some Terms for Elections

- Race
  - An element on a ballot that people vote on
- Overvote
  - More votes cast by a voter in a particular race than is allowed for a voter
- Undervote
  - Fewer votes cast by a voter in a particular race than is allowed for a voter
- Example
  - Race is 3 open seats for city council, 5 candidates for those seats
  - I vote for 2 of them, not 3: that’s an undervote and it counts
  - I vote for 4 of them, not 3: that’s an overvote and it doesn’t count

How an Election Works in Yolo County, CA

- Voters:
  - Go to voting center (precinct), give name
  - Get ballot, enter booth, vote using marker to mark ballot
  - Put ballot in protective sleeve, leave booth
  - Drop ballot into ballot box
    - If provisional or conditional, put ballot and sleeve into envelope with voter's name, reason for the challenge (provisional) or condition (conditional) on the *outside*
- Vote-by-mail voters:
  - Fill in ballot
  - Put ballot into inner envelope
  - Put inner envelope into mailing envelope; sign the *outside* and mail it in

End of the Day

- Election officials take ballot box to County seat
- Election officials remove ballots from envelopes
  - Provisional and conditional ballots handled separately
- Ballots counted, put into bags marked with precinct and count
- Ballots removed from bag, run through automatic counters
  - Humans intervene when problems arise
  - Intermediate tallies written onto flash cards
  - Every so often, cards removed, walked to tally computer, inserted, votes counted
- Reported tallies periodically updated, given for posting to web

And Then . . .

- All places have provisional ballots
    - These are cast when it is unclear if the person is allowed to vote
    - In California, ***always*** on paper, never electronic
- California allows conditional ballots
    - These are cast by folks who register at the election (same day registration)
- Conditional and provisional ballots must be validated before being counted
- California also allows mail-in ballots arriving up to 3 days after Election Day to be counted

The Canvass

Required by California law:

- Ballots for 1% of precincts counted by hand
  - Chosen with throw of dice; if some races not in precincts selected, add more in until all covered
  - Some counties have legal authority to use risk-limiting audit as well or instead
  - In California, you *must* use paper for this (hence, all DREs have VVPATs)
- Compared to tallies from election
  - If different, must be reconciled
- Certify final counts to Secretary of State
  - Has to be done within some number of days after election

Some Election Requirements

- Voter validation (authenticated, registered, has not yet voted)
- Ballot validation (voter uses right ballot, results of marking capture intent of voter as required by law)
- Voter privacy, secrecy (no association between voter, ballot; includes preventing voter showing others how he/she voted)
- Integrity (ballots unchanged, votes tallied accurately)

Some Election Requirements

- Voting availability (voter must be able to vote, materials must be available)
- Voting reliability (voting mechanisms must work, even under adverse circumstances)
- Election manageability (process must be usable by those involved, including poll workers)
- Election transparency (audit election process, verify everything done right)

What Should an E-Voting System Do?

- Replace manual activity, existing technology used in election process with better technology
  - Better in the sense of improving some aspect of the election process
- Examples
  - Easier to program ballots than print them
  - Can handle multiple languages easily
  - Easier to tally than hand counting

Assurance

- Provide sufficient evidence of assurance to target audience that using e-voting systems makes elections at least as secure, accurate, etc. as elections without them (that is, using paper ballots)
- Who is “target audience”?
  - Computer scientists, election officials, politicians, *average person*

A Smattering of Problems

- Boone County, IN, 2003: 144,000 votes cast in a county with about 6,000 voters
- In 2006, polls opened late in several California (CA) counties (San Diego, Alameda, Plumas, Kern, Solano) due to system problems
- South Bronx, NY, 2010: a scanner miscounted 69/103 (70%) of ballots in Sep., then 156/289 (54%) in Nov.
- Los Angeles, CA, 2020: electronic poll books had connectivity problems, resulting in unacceptably long lines; BMDs failed, had paper jams

Results of Testing

- 2003: Johns Hopkins people analyzed voting system program
- 2004: RABA rigged mock election in about 30 minutes
- 2006: Florida CD-13 election *post mortem*
- 2007: California Top-to-Bottom Review, Ohio EVEREST review
- 2011: Washington DC internet voting test compromised
    - And the friendly attackers threw out the hostile ones
- 2014: Analysis of Estonia e-voting systems: many vulnerabilities found
- 2020: Voatz mobile voting app based on “blockchain technology”: many vulnerabilities found

How to Get Better

- You need both standards and testing
- They must be independent of the developers of the systems
- They need to consider the users, operators, and maintainers of the systems
- Reports should show what tested, why, and how
- For e-voting systems, penetration testing is a *must*

Add in the Internet

- It will enable authorized voters who cannot vote due to distance (or other factors) to do so
- It will increase authorized voter participation
- It will bring our elections into the modern, technological world
- It will be cheaper because we don’t have to store the paper ballots

Problem:

- Election systems are now accessible to many more people than authorized voters!

Where Would Attackers Strike?

- Probably not regular, individual electronic voting systems
- But attack the vendors and change the programs that run on those systems, or on the tallying systems
- Or hit the voter registration databases to disenfranchise voters

Remote Voter Verification of Ballots

- Trick here is to protect against the validating mechanism being corrupted
- Example: we examined a system that enabled voters to check that their ballots were recorded correctly, and counted correctly, remotely
  - Used very neat cryptography, done by experts
  - We simply changed the web page on which the information that the user used to do the validation – no cryptography involved!

Moral: attackers don’t have to rig or corrupt an election

They just have to make you *think* they did!

Blockchains

- Background
  - Take ballot or chain of ballots and compute a hash from them
  - Encrypt this with a cryptographic key you keep secret (private key)
  - Publish the inverse cryptographic key (public key) so others can verify the small value was not changed
- For voting: many proposals for handling the chains

Why Blockchains Fail for Elections

- Problem #1: denial of service (already discussed)
- Problem #2: how are those cryptographic keys generated?
  - A. Voter generates the pair (this is how it’s usually done for other uses), and publishes the public key
  - A′. I vote multiple times, possibly under the name of a different voter each time. Prove I was the one who did this, and determine which votes are mine.
  - A′′. I want to sell my vote. I give my private key to the purchaser. She can use the public key to verify that is my private key, and then see how I voted by finding the specific ballot added using that public key.
  - B. Election officials assign key. Now *they* can determine how I voted!

How Not to Test Voting Over the Internet

- Occasional bills in various legislatures to do a “pilot study” using Internet voting in a real election
- A valid test requires knowing “ground truth”, that is, what the results of the election should be
- How do you know this in a real election?

Conclusion of Course

- Thank you for being such good students
- Please remember to fill out the evaluations
  - Let me know if I could do something better (suggestions about how are also welcome!)
  - Let me know if I did something well, so I don't change it

Parting Thought

Man has always assumed that he is more intelligent than dolphins because he has achieved so much—the wheel, New York, wars and so on—while all the dolphins had ever done was muck about in the water having a good time.

But, conversely, the dolphins had always believed that they were far more intelligent than man—for precisely the same reasons.

— D. Adams, *Hitchhiker’s Guide to the Galaxy*