Towards Repeatable and Reproducible Assurance for Modern PPs

Edward Morris

September 11, 2014

Gossamer Laboratories

Attendees of Yesterday's Talk?

*Gossamer Legal Disclaimer: actual checks have no cash value and will not be honored.

As before, please direct feedback to Tammy Compton:
TammyCompton@GossamerSec.com +1 (240) 994-9770

Agenda

- Review of modern CCEVS/NIAP Protection Profiles
- Strengths and Challenges for Assurance through Testing
- Development of Assurance Testing Tools and Suites
- Possible Solutions
- Recommendations

Modern CCEVS/NIAP PPs

Modern PPs loosely defined as:

- EAL0
- CC version 3.1
- Focus on correct cryptography and entropy
- No source code review required
- Protocol-level evaluator testing required

22 Current, Modern PPs exist

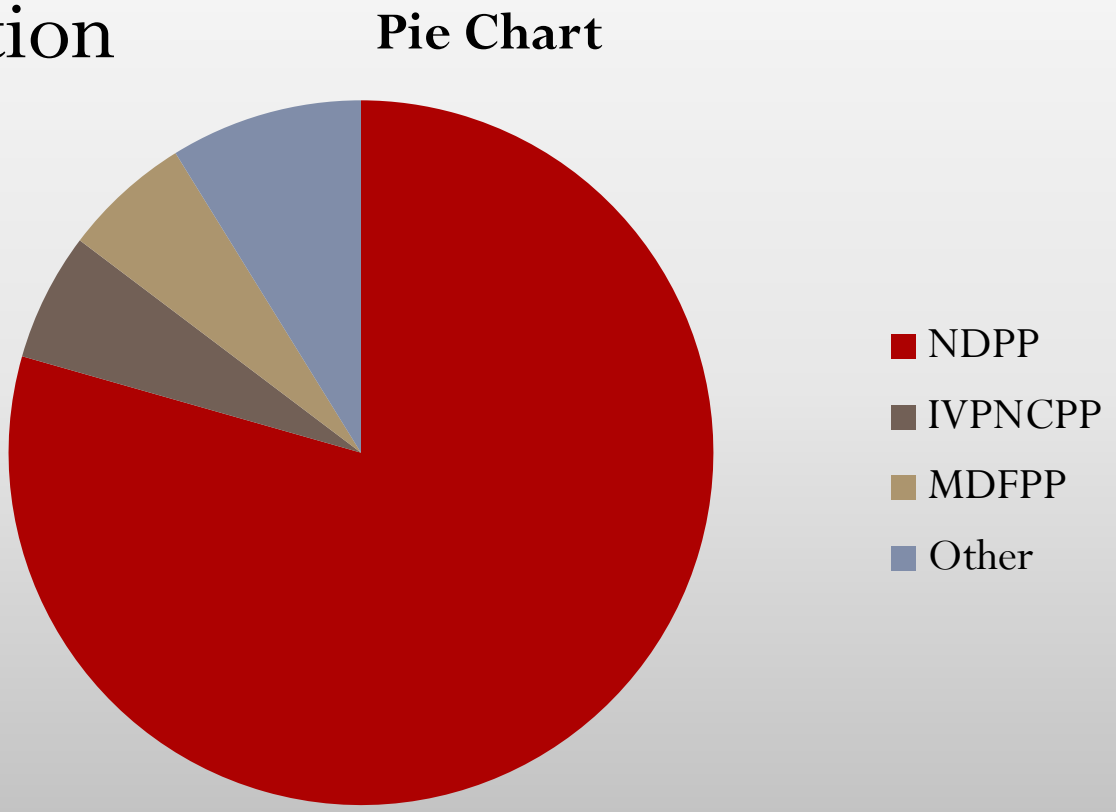
Modern CCEVS/NIAP PPs

Number of products tested against Modern PPs

- Network Device PP (27 of 34 products listed) including
  - Traffic Filtering Firewall EP (4) and
  - IPsec VPN GW (1)
- IPsec VPN Client (2)
- Mobile Devices (2)
- Other: ESM (1), VoIP (1), SWFDE (1),

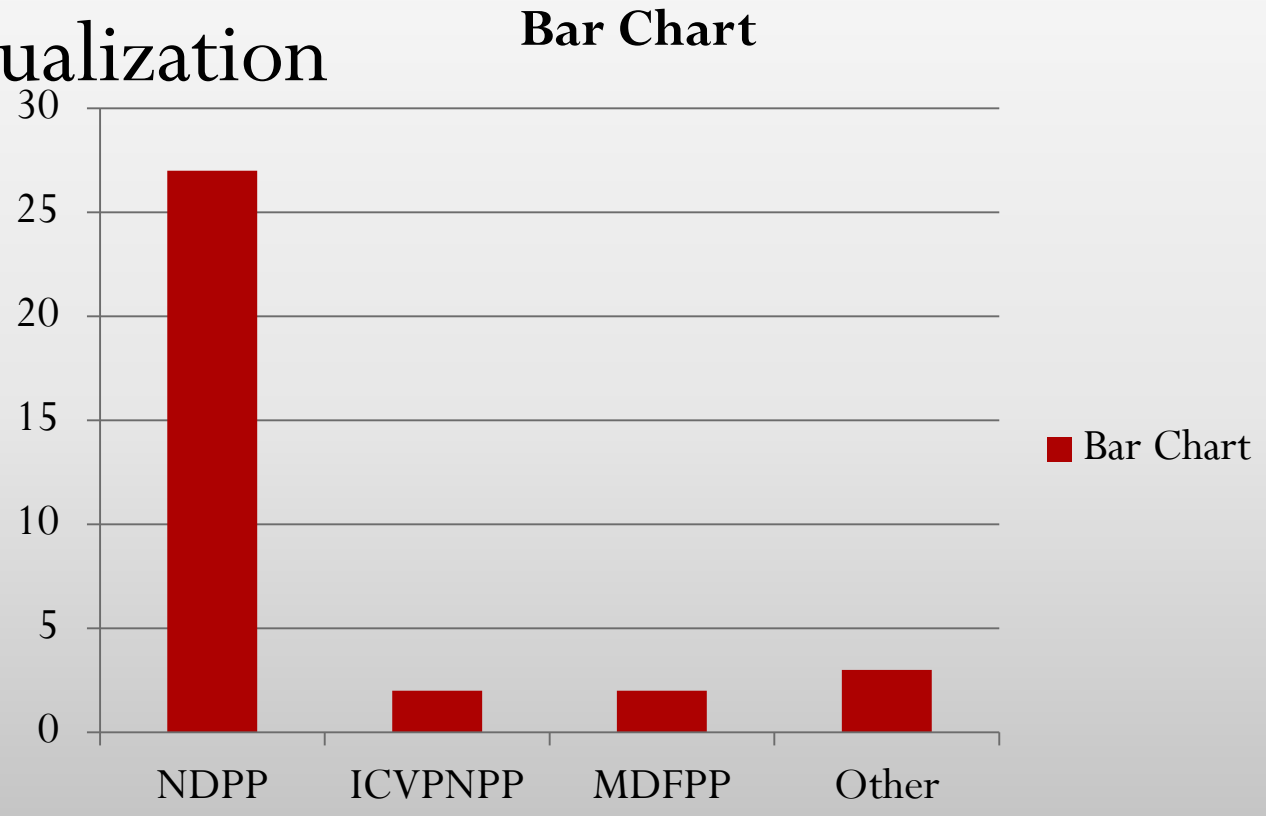
Modern CCEVS/NIAP PPs

Visualization



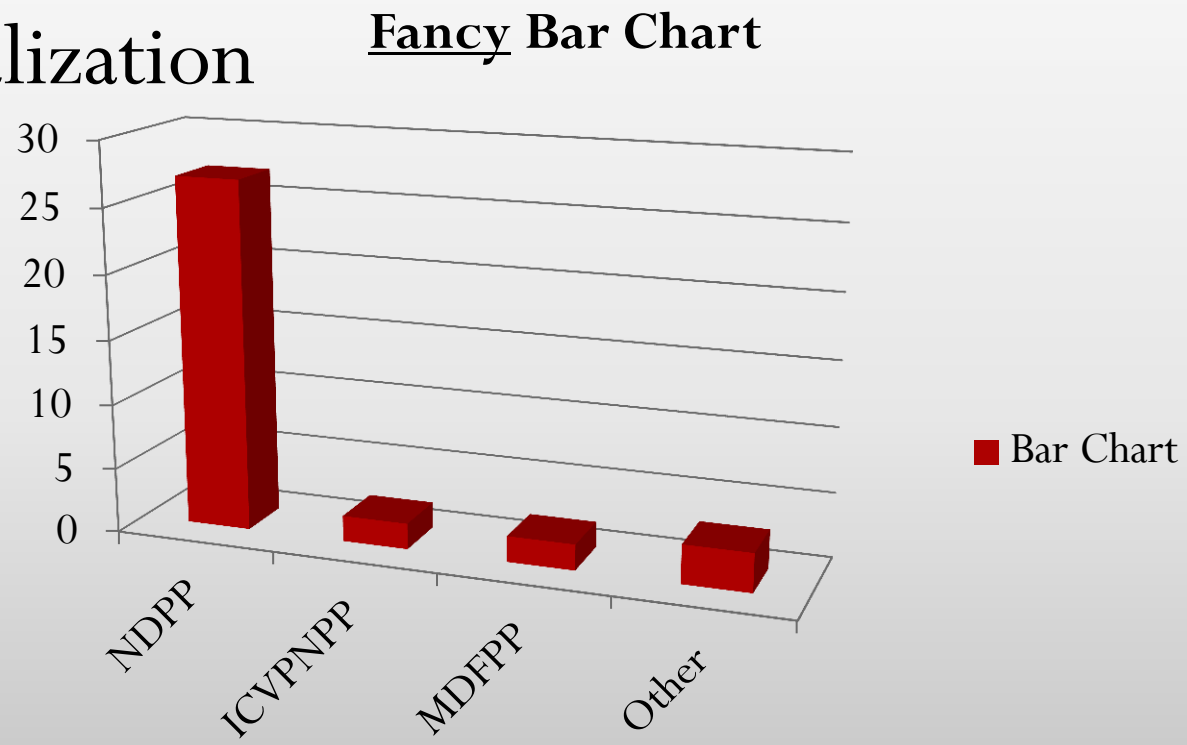
Modern CCEVS/NIAP PPs

Visualization



Modern CCEVS/NIAP PPs

Visualization



Modern CCEVS/NIAP PPs

Examples of new requirements

- Cryptographic algorithm testing
  - NIST/CAVP algorithms with
  - Some augmentation
- Focus on industry protocols:
  - IPsec
  - SSH
  - TLS/HTTPS

Modern CCEVS/NIAP PPs

Test Example No. 1 (NDPP v1.0)

- SFR: FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*assignment: number of bytes*] bytes in an SSH transport connection are dropped.

- AA: Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

Modern CCEVS/NIAP PPs

Test Example No. 2 (NDPP v1.1)

- SFR: FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites: (list of mandatory & optional suites)

- AA: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement.

Modern CCEVS/NIAP PPs

Test Example No. 3 (MDFPP v1.1)

- SFR: FCS_TLS_EXT.1.1
- AA: …Test 5: 4th bullet: "Modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field must not be the CA used to sign the client's certificate. The evaluator shall verify that the server rejects the connection after receiving the Client Finished handshake message."

Strengths and Challenges for Assurance through Testing

The strengths of assurance through testing

- Intended as objective tests - Result is pass/fail (e.g., product drops packet or terminates/refuses connection)
- Tests the fielded product (*in vivo*)
- No access to source code required ("black-box")
- Presumably checks for common deficiencies/weaknesses in protocol implementations
- Better efficiencies of testing effort and product security improvement

Strengths and Challenges for Assurance through Testing

Challenges for assurance through testing

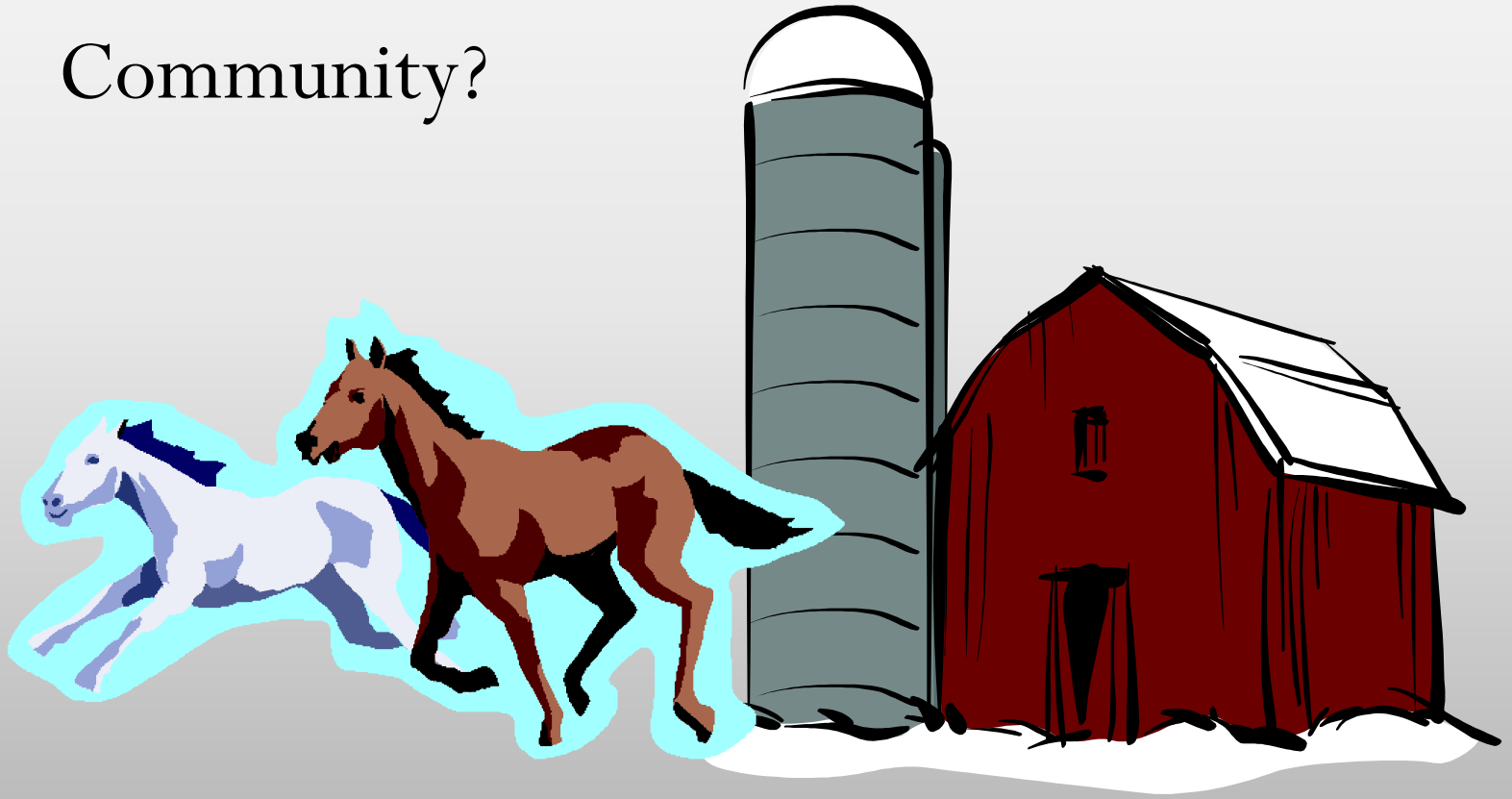
- Requirements are heavy in cryptography
- Evaluators must be well-versed in the low-level details of the different protocols
- Tests must be well specified
- ***No tools or test suites currently available!***

Development of Assurance Testing Tools and Suites

- Who should author the testing tools?
  - Validation Schemes?
  - International Technical Communities?
  - Evaluation Laboratories?
  - Product Vendors?
  - 3rd Commercial Testing Tool Vendors?

Development of Assurance Testing Tools and Suites

- Validation Scheme or International Technical Community?



Development of Assurance Testing Tools and Suites

- Evaluation Laboratories?

Development of Assurance Testing Tools and Suites

- Product Vendors?

Development of Assurance Testing Tools and Suites

- 3rd Party Commercial Testing Tool Vendors?

Development of Assurance Testing Tools and Suites

- So who should author the testing tools?
- It actually doesn't matter, if the tool is
  - Transparent (tool source available) and freely available OR
  - The tool outputs conclusive data supporting all results ("Raw Output")
- But either condition has problems

Problems with Open, Freely Available Tools

- There's little incentive to invest one's own effort developing a tool that benefits others.
  - Laboratories likely to view their tools as a competitive advantage.
  - Vendors are likely to have a similar view
  - "Freely available" won't work for 3rd Party commercial testing tool vendors
- So the economic incentives don't align to "organically" spur open tool development.

Problems with Raw Output Tools

- Many of these tests are intricate tests that don't lend themselves to cursory examination.
  - An X.509 CA cert in the cert-chain with an invalid path length might be difficult to detect, even if one were provided with all the certificates in the test PKI.
  - Reviewing the TLS handshake exchange packets to detect a corruption to the nonce is not easy.
- So tools with Raw Output burden evaluations and don't scale well.

Testing Tools (*de facto*) Solution 1

- Given that vendor and laboratories have already created tools,
  - require evaluations to produce and submit "raw output" evidence (packet captures and/or log files) to substantiate their conclusions.

- Pros:
  - Easy to implement, no burden on the oversight scheme to require this from labs
  - Produces (hopefully) objective results that can be shared with others
- Cons:
  - Burdens labs with generation, collation, and reporting of raw evidence
  - Burdens oversight and inter-scheme consistency processes with reviewing packet captures and logs (a tedious, manual process)
  - Even with packet captures, some tests still might be unclear without having access to the tools and tool source

Testing Tools Solution 2

- Given that vendor and laboratories have already created tools,
  - Require that evaluations first demonstrate the veracity of the results/conclusions produced by the tool.

- Pros:
  - Ensures laboratory proficiency with subject matter.
  - Once tool veracity is established, evaluators can document and oversight bodies can review only the high-level results.
  - Produces (hopefully) objective results that can be shared with others
- Cons:
  - Hard to demonstrate tool validity (chicken and egg)
  - Requires development of testing artifacts to test the testing tools—effectively equal to the work of developing tools!

Testing Tools Solution 3

- Collaborative development of transparent, freely available tools.

- Pros:
  - Can be freely distributed and collectively reviewed, breeding international confidence in the tool and evaluation conclusions reached using it.
  - Greatly reduces the evaluator, scheme, and inter-scheme workload (cursory review of the high-level results should suffice).
- Cons:
  - Developing a good tool is difficult work
  - Getting participation and agreement on development of a low-level testing tool may prove too difficult
  - Can have the effect of diminishing evaluator proficiency

Testing Tool Recommendations

- Implement Solution 3 if possible: iTC's should develop either testing artifacts or testing tools
- Alternatively (or in the interim, continue to) use Solution 1: evaluations continue to provide "raw data" to substantiate results.
- Explore creative ways to incentivize development contribution or to amortize development costs, e.g.
  - mutual recognition applies only if the host scheme first contributes development resources.
  - development "tax" levied upon PP evaluations (passes a cost to vendors commensurate with their usage of the PP).
  - other ways?

Questions?

Contacts:

- Ed Morris
  - EdMorris@GossamerSec.com

www.gossamersec.com
www.facebook.com/gossamersec
@gossamersec