

Redefining Security Criteria for Networking Devices with Case Studies

Ying-Dar Lin, Chia-Yin Lee, and Hao-Chuan Tsai | National Chiao Tung University

Common Criteria, ICSA Labs, and NSS Labs—three well-known standard security criteria—combine with the RealFlow stability test to form a set of lightweight total security criteria, providing wide coverage on documentation, security functionality, performance, self-protection, and stability in a short evaluation period.

To improve the security quality of information and communications technology (ICT) products sold in Taiwan, we worked with Taiwan's National Communications Commission to establish security criteria.¹ To meet the security criteria, vendors or developers often need to enhance their products, which in turn improves overall security quality. However, defining meaningful security criteria is challenging because many aspects must be considered during a short evaluation period.

Common Criteria for Information Technology Security Evaluation (CC; ISO/IEC 15408) offers precise methodologies to ensure the security functionality of ICT products.²⁻⁴ CC helps developers deal with security requirements during the entire product development cycle. However, it has two major drawbacks: it's limited to document review of developer-provided scenarios, and evaluation times are lengthy.

Two independent test organizations—ICSA Labs (<https://www.icsalabs.com>) and NSS Labs (<https://www.nsslabs.com>)—evaluate product quality using test methodologies rather than documents generated in the product development cycle. ICSA emphasizes detecting malicious traffic, whereas NSS focuses on performance and self-protection. These test methodologies could overcome CC's drawbacks and check the blind spots in document review. However, several security products that passed either the ICSA or NSS test failed

in real-life attacks. In addition, ICSA and NSS don't test stability using real traffic.

Real traffic testing is much more powerful in ensuring networking device stability; artificial traffic merely emulates and covers a small portion of protocol messages and parameters. The complexity of real traffic could trigger a product's defects, possibly disabling its security functionalities. To test for stability, we developed a real traffic test called RealFlow, which explores more program execution paths in devices under test (DUTs) and triggers more defects that would otherwise be found by customers.^{5,6}

Combining CC, ICSA, NSS, and RealFlow to cover all aspects of security results in a large criteria with a long evaluation period and hence isn't an optimal solution. Instead, we adopt a best-of-breed strategy to establish a set of lightweight total security criteria. We selected CC document reviews that affect product quality, the most important and efficient ICSA and NSS test cases, and RealFlow with a shorter test period.

In this article, we review CC, ICSA, and NSS methodologies; illustrate the coverage of the best-of-breed criteria; and describe the results of a pilot run.

Existing Security Evaluation for ICT Products

Let's start by describing CC, ICSA, and NSS methodologies briefly and pointing out their drawbacks.

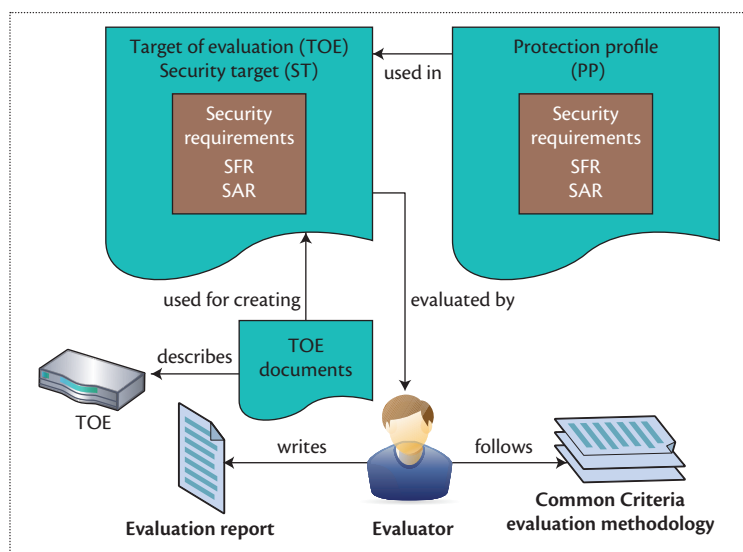


Figure 1. The Common Criteria evaluation process. The evaluator verifies how the proposed security target of the target of evaluation conforms to the relevant protection profile and which evaluation assurance level the developer chooses to test against. Finally, the evaluator writes the evaluation report.

Common Criteria: Mostly Document Review

CC evaluates ICT product security in three phases: document preparation, evaluation, and conclusion.⁷

In the document preparation phase, a protection profile (PP) is proposed by a demander or community. The primary goal of PP is to establish security objectives, assumptions, security functionality requirements (SFRs), and security assurance requirements (SARs). An SFR describes individual security functions that a product should provide, and an SAR measures and evaluates the product to ensure SFR compliance. PPs also specify generic security evaluation criteria to substantiate the developer's claim of a given target of evaluation (TOE). Note that a TOE involves the DUT and related guidance. Typically, PPs specify seven evaluation assurance levels (EALs) that indicate the evaluation's depth and rigor. A security target (ST), established by developers, defines security requirements for a given TOE. Because the ST is a complete and rigorous description of security coverage, developers must refer to one or multiple PPs to propose a specific ST for future evaluation.

In the evaluation phase, each EAL corresponds to a package of SARs, which covers complete product development with a given level of strictness. Figure 1 illustrates the evaluation process and the relationships among PP, SFR, SAR, TOE, and ST. The evaluator initially verifies how the proposed ST conforms to the relevant PP and the EAL that the developer chose. By following the TOE document, the evaluator generates test cases to verify the TOE's security features and

then determines how well the TOE satisfies the SFRs defined in the ST.

In the conclusion phase, the evaluator provides an evaluation report to the validator (a certification body), which generates a validation report for publication.

CC claims to ensure the quality of the entire product development cycle; however, as we mentioned, it has some drawbacks. For example, verifying a TOE with EAL 4—a moderate evaluation level—would take 12 to 16 months of evaluation and testing with iterative phases. This might be problematic because ICT products usually have short life cycles. In addition, if a product is updated with a new version during evaluation, it must be reevaluated. Another drawback is that developers propose most ST test plans. Because developers produce the required documents, they could choose to produce only the parts they specialize in. For example, a developer might design an intrusion detection and prevention system (IDP) and list only HTTP responses as the major requirement. However, this is inadequate because IDPs should handle other flows as well.

To meet the security requirements of products with a short life cycle, lab or field tests are essential. Many test labs, such as ICSA and NSS, provide such services, but most focus on validation against minimum standards of DUTs without a full-spectrum analysis.

ICSA: False Positives and False Negatives

ICSA provides several test methodologies:⁸

- *administration* covers the capacity of remote DUT administration;
- *identification and authentication* verifies the capacity of the DUT that requires and enforces user identification followed by authentication with passwords;
- *traffic flow* requests that the DUT pass all benign IP traffic (up to 80 percent of the maximum throughput) according to the established policy;
- *logging* records all the required log events, such as attempts to sneak attacks through the DUT;
- *functional testing* inspects administrative capabilities; and
- *security testing* verifies the self-protection capability.

A review of the ICSA shows that the key to passing the criteria is the test on the false negative and false positive (FN/FP) rates for malicious and benign traffic. For example, ICSA verifies an IDP's FN/FP using a set of contemporary and core vulnerabilities. The former is no older than one year as of the date of the vulnerability set, whereas the latter is older than one year. The vulnerability set published on 1 September 2011 contains 35 sample contemporary vulnerabilities and 88 sample core vulnerabilities. All contemporary vulnerability samples must

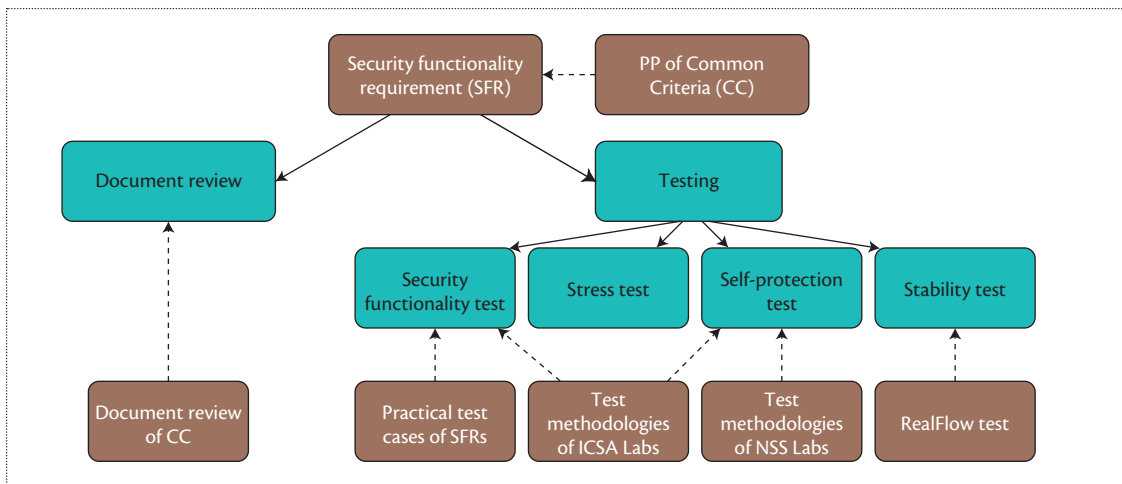


Figure 2. Framework of the proposed security criteria. We adopt the best-of-breed strategy to select the criteria from Common Criteria, ICSA Labs, and NSS Labs. The resulting criterion provides a wider coverage on documentation, security functionality, performance, self-protection, and stability.

be detected, and only 10 percent of core vulnerabilities samples can have FN. FPs are unacceptable.

NSS: Self-Protection and Performance

To address ICT products' increasing complexity, NSS provides security effectiveness and performance tests.¹¹ The security effectiveness test cases include verifying DUT security by live exploits—for instance, utilizing packet and stream fragmentation to assess the fragment reassembly mechanism, employing random URL encoding techniques to determine whether DUTs can block obfuscated URLs, and determining whether DUTs could be evaded by inserting additional spaces and telnet control sequences in the FTP commands. Security effectiveness tests focus on self-protection test cases—live exploits such as buffer overflow, code injection, cross-site scripting, directory traversal, and privilege escalation—to verify DUT protection capabilities. NSS doesn't establish explicit DUT FP/FN criteria.

NSS performance tests attempt to find the highest throughput under various conditions while running security functions. Performance test cases include utilizing UDP packets of varying sizes to measure raw packet processing performance, including testing the connection dynamics (with excessive concurrent TCP connections, excessive response time for HTTP transactions/Simple Mail-Transfer Protocol [SMTP] sessions, and unsuccessful HTTP transactions/SMTP sessions); estimating theoretical maximum concurrent TCP connections and maximum HTTP connections; verifying the DUT's ability to preserve state across many open connections over an extended time period; testing the HTTP capacity with transaction delays; and simulating real-world traffic to test the DUT's

maximum throughput. For example, to determine an IDP's raw packet processing capability, NSS uses varying packet sizes, such as 128, 256, 512, 1,024, and 1,514 bytes, with traffic loads of 25, 50, 75, and 100 percent of IDP maximum throughput. In addition, it generates specific background traffic to simulate the real-world environment. The background traffic is made up of 33 percent HTTP text, 18 percent HTTP images, 18 percent SMTP, 8 percent HTTP videos, 8 percent FTP, 6 percent DNS, 4 percent Secure Shell (SSH), 3 percent AOL instant messages, 1 percent Session Initiation Protocol (SIP)/Real-Time Transfer Protocol (RTP), and 1 percent BitTorrent.

Lightweight Total Security Criteria

Again, simply combining CC, ICSA, and NSS to maximize coverage isn't feasible due to the prolonged evaluation period. Only parts of CC documents check end product quality, whereas ICSA focuses on the security functionality test and FN/FP, and NSS focuses on self-protection and performance. Because none cover the important aspect of stability, which might also pose security threats, we use our RealFlow criteria.⁶ To limit the evaluation period to two months, we adopted a best-of-breed strategy to select the criteria, as Figure 2 shows. Although DUT-provided SFRs can be described in the review document, using test cases to ensure SFR validity is essential. Our framework consists of document review and testing; the former is solely from CC and the latter contains SFRs of CC, ICSA, NSS, and RealFlow test cases. We renamed *performance* as *stress* to emphasize the maximum performance with security functions on. Each component in the framework impacts security if a DUT fails to pass it.

Table 1. Our proposed security criteria versus Common Criteria.

| Assurance class | Assurance component | Assurance component description | Protection profile/ Common Criteria | Basic | Advanced |
|--------------------------|---------------------|---|-------------------------------------|--|----------|
| Development | ADV_ARC.1 | Architectural design with domain separation and non-bypassability | ■ | ■ | ■ |
| | ADV_FSP.2 | Security-enforcing functional specification | ■ | ■ | |
| | ADV_FSP.4 | Complete functional specification | | | ■ |
| | ADV_TDS.1 | Basic design | ■ | | |
| | ADV_TDS.2 | Architectural design | | ■ | ■ |
| Guidance documents | AGD_OPE.1 | Operational user guidance | ■ | ■ | ■ |
| | AGD_PRE.1 | Preparative user guidance | ■ | ■ | ■ |
| Life cycle support | ALC_CMC.1 | Labeling of the target of evaluation (TOE) | | ■ | ■ |
| | ALC_CMC.2 | Use of a configuration management system | ■ | | |
| | ALC_CMS.2 | Parts of the TOE configuration management coverage | ■ | ■ | ■ |
| | ALC_DEL.1 | Delivery procedures | ■ | | |
| | ALC_FLR.2 | Flaw reporting procedures | ■ | | |
| Tests | ATE_COV.1 | Evidence of coverage | ■ | Covered by document review and testing | |
| | ATE_FUN.1 | Functional testing | ■ | | |
| | ATE_IND.2 | Independent testing sample | ■ | | |
| Vulnerability assessment | AVA_VAN.2 | Vulnerability analysis | ■ | Covered by self-protection test | |

Document Review: EAL 2+

We used existing PPs’ SFRs to produce the document review evaluation items. Table 1 shows the differences between the existing CC PPs and the proposed basic and advanced security criteria. SFR assurance is categorized into five classes, each with several assurance components corresponding to EAL. In general, assurance components for the higher EAL also include those defined in the lower EAL. For example, the component ADV_ARC.1, denoting the evaluation of the architectural design development documents, is contained in EAL 2 to EAL 4. The component ADV_FSP.4, denoting the complete functional specification, is contained only in EAL 4.

Compared to a PP with EAL 2, the proposed security criteria achieve at least all the assurance components of EAL 2 (for basic criteria) and partial assurance components of EAL 4 (for advanced criteria) and can be categorized as EAL 2+. In addition, CC specifies the TOE’s life cycle support using a configuration management system and delivery procedure. Our criteria adopt only partial components of life cycle support because the evaluation is time-consuming and product life cycle

support could be evaluated by lab and field tests to obtain reasonable estimations.

Selecting ICSA and NSS Test Cases by the Impact Metric

According to the essential TOE SFRs, we referred to the ICSA and NSS test methodologies to design the security functionality, stress, and self-protection test cases. We excluded the nonapplicable items from the ICSA and NSS test cases, then selected from the remaining test cases using two factors: the importance of the test case and the test’s expected manpower. We ranked importance as high, medium, and low, with scores of 10, 5, and 1, respectively, and used man-day as the unit of measurement for the expected manpower. Then, we normalized the impact *I* of each test case as $I = score/man\text{-}day$. Finally, we set a minimum threshold *T* to determine whether to pick the test case.

We use the coverage of the IDP test methodologies as an example to illustrate which test cases we selected from ICSA and NSS, as Tables 2 and 3 show. We also increased the selected tests’ diversity and practicability.

Table 2. ICSA test cases for intrusion detection and prevention systems (IDPs).

| Main class | Test case | Importance | Expected man-day | Covered by our criteria |
|-----------------------------------|--|------------|------------------|--|
| Administration | Remote administration | High | 0.5 day | Security functionality test (secure administrative interface) |
| Identification and authentication | Authenticate to administrative function | High | 0.5 day | Security functionality test (secure administrative interface) |
| | Strength of password (optional) | Medium | 1 day | N/A |
| Traffic flow | Passing IP traffic | High | 1.5 days | Security functionality test (false negative and false positive [FN/FP] test) |
| Logging | Required log events | High | 0.5 day | Security functionality test (security events records) |
| | Required log data | High | 0.5 day | Security functionality test (security events records) |
| | Required data presentation | High | 0.5 day | Security functionality test (security events records) |
| | Linking multiple logs (conditional) | Medium | 1 day | N/A |
| Functional testing | Administrative functions work properly | Medium | 0.5 day | Security functionality test (secure administrative interface) |
| | Average one-way latency | Low | 1 day | N/A |
| Security testing | System under test (SUT) not addressable | High | 0.5 day | Security functionality test (secure administrative interface) |
| | No unauthorized access to administrative functions | High | 1 day | Security functionality test (evasion detection) |
| | Engine not vulnerable | High | 1.5 days | Self-protection test (remote management for abnormal flows) |
| | Coverage of attacks against relevant vulnerabilities | High | 1.5 days | Self-protection test (remote management for abnormal flows) |
| | Coverage of trivial denial-of-service (DoS) attacks | High | 1.5 days | Self-protection test (prevention of DoS attacks) |
| | Repeated protection | Medium | 1.5 days | N/A |
| | No false positives after tuning | High | 1.5 days | Security functionality test (FN/FP test) |

For example, we used Common Vulnerabilities and Exposures to evaluate IDPs' security functionalities, employing exploits with a score greater than or equal to 7.0—high risk in the Common Vulnerability Scoring System—to perform the FN test.

As Table 2 shows, certain test cases, such as average one-way latency, weren't adopted because their principal parts were already covered by the stress test. From the results in Tables 2 and 3, we dropped half the ICSA functional test cases. By comparison, only system exposure from NSS wasn't adopted. Most attackers utilize service exploits instead of system exploits, so because of its low importance and high expected manpower, we discarded the system exposure. Tables 4 and 5 show our

proposed test cases' coverage percentage compared to ICSA and NSS.

RealFlow for Stability

Although NSS claims it utilizes real-world traffic for testing, it actually evaluates DUTs with artificially emulated flows that have specific and fixed patterns of packet traces. These scenarios are inadequate because network environments have much greater multiplicity and complexity, including various protocol messages and parameters.

RealFlow utilizes real traffic from real networks.⁶ It consists of a campus beta site for live testing and a packet library (PCAP Lib) for replay testing.¹⁰ Live

Table 3. NSS test cases for IDPs.

| Main class | Test case | Importance | Expected man-day | Covered by our criteria |
|---------------------|---|------------|------------------|---|
| Detection engine | System exposure | Medium | 1 day | N/A |
| | Service exposure | High | 1 day | Self-protection test (prevention of DoS attacks) |
| Coverage by result | Arbitrary code execution | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | Buffer overflow | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | Code injection | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | Cross-site script | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | Directory traversal | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | Privilege escalation | High | 0.5 day | Self-protection test (remote management for abnormal flows) |
| Evasion | Unmodified exploit validation | High | 0.5 day | Security functionality test (evasion detection) |
| Fragment reassembly | Packet fragmentation | High | 1 day | Stress test (throughput test) |
| | Stream segmentation | High | 1 day | Stress test (throughput test) |
| | Remote procedure call (RPC) fragmentation | High | 1 day | Stress test (throughput test) |
| | URL obfuscation | Medium | 0.5 day | Self-protection test (remote management for abnormal flows) |
| | FTP evasion | High | 1 day | Security functionality test (evasion detection) |
| Performance | Raw packet processing (UDP traffic) | High | 1 day | Stress test (throughput test) |
| | Connection dynamics | High | 1 day | Stress test (maximum number of connection)* |
| | Behavior of the state engine under load | High | 1 day | Stress test (maximum connection rate)* |
| | HTTP capacity with no transaction delays | Medium | 0.5 day | Stress test (maximum connection rate) |
| | HTTP capacity with transaction delays | Medium | 0.5 day | Stress test (maximum connection rate) |
| | Real-world traffic | High | 1 day | Stress test (throughput test) |

* For advanced security criteria only

testing is done by deploying DUTs in the beta site to carry real traffic. If a DUT fails, the beta site detects and bypasses it to recover network connectivity. To achieve DUT failures' reproducibility, replay testing captures and classifies real traffic in the PCAP Lib and later replays the traffic onto DUTs in the lab instead of the beta site. When defining our security criteria, we adopted replay testing using the same packet traces captured monthly for the purposes of reproducibility and

fairness. However, the campus beta site has some deficiencies. For example, the network applications on the campus beta site aren't the same as those on an enterprise network.

Previous research shows that RealFlow is good for finding stability and compatibility defects.⁶ For example, one antimalware product being tested had compatibility defects. Many Web TV applications, such as Ipobar, TVant, and Vigor, were incompatible with the

Table 4. Proposed security criteria test case coverage compared to ICSA.

| Main class | Coverage | | Coverage (%) | |
|-----------------------------------|----------|----------|--------------|----------|
| | Basic | Advanced | Basic | Advanced |
| Administration | 1/1 | 1/1 | 100 | 100 |
| Identification and authentication | 1/2 | 1/2 | 50 | 50 |
| Traffic flow | 1/1 | 1/1 | 100 | 100 |
| Logging | 3/4 | 3/4 | 75 | 75 |
| Functional testing | 1/2 | 1/2 | 50 | 50 |
| Security testing | 6/7 | 6/7 | 85.7 | 85.7 |

Table 5. Proposed security criteria test case coverage compared to NSS.

| Main class | Coverage | | Coverage (%) | |
|---------------------|----------|----------|--------------|----------|
| | Basic | Advanced | Basic | Advanced |
| Detection engine | 1/2 | 1/2 | 50 | 50 |
| Coverage by results | 6/6 | 6/6 | 100 | 100 |
| Evasion | 1/1 | 1/1 | 100 | 100 |
| Fragment reassembly | 6/6 | 6/6 | 100 | 100 |
| Performance | 4/6 | 6/6 | 66.7 | 100 |

antimalware solutions. After our analysis, we found the incompatibility was caused by defects in the software agent. Specifically, the software agent conflicted with the multicast stream process in the operating system as well as with many security and game applications, which were implemented with a protection mechanism that prevented the software agent from hooking them with dynamically linked library injections. Such defects are almost impossible to trigger with artificial traffic.

The period starting from the beginning of a test until the next defect is found is called *time to fail*. If the time to fail exceeds four weeks, it's considered converged. We found that all products' time to fail increases iteratively as developers fix their defects, meaning that product quality improves through iterative RealFlow testing. To reduce our security criteria's evaluation period, we decreased the passing threshold from four weeks to one week for the basic level and two weeks for the advanced level.

Criteria for Eight Product Categories

We used the above strategies to establish lightweight total security criteria for eight product categories: firewalls, IDPs, antivirus gateways, antispam systems, Web application firewalls (WAFs), layer-2 switches, layer-3 switches, and application control systems. Note that application control systems protect managed hosts and servers by allowing or denying network application use based on administrator-established policies. The evaluation items of our criteria among

different product categories are approximately 70 percent identical. The remaining 30 percent are associated with various DUT functionalities. Most of the different items belong to security functionality testing and self-protection testing. In Tables 6 and 7, we summarize the evaluation criteria of security functionality and self-protection testing for firewalls and IDPs to demonstrate their diversity.

To propose security criteria for a specific product category, we surveyed related PPs to select essential SFRs. We referred to the first-tier products' operation guides and surveyed technical reports to propose practical test cases corresponding to those SFRs for security functionality testing. We also referred to ICESA's test methodologies to design the criteria for each test case. Similarly, for self-protection tests, we surveyed related literature to understand possible product vulnerabilities and then proposed the necessary test cases. We then referred to both ICESA and NSS test methodologies to design the criteria for each test case.

Case Studies

Here, we illustrate the proposed security criteria's pilot run and discuss the test results. DUT developers must establish an ST to define all met security requirements for a TOE, providing three essential tables in the document review phase—security functionality specification, subsystem description and classification, and security architecture description—along with

Table 6. Evaluation criteria of security functionality and self-protection tests—firewalls.

| Class | Test case | Criteria | Basic | Advanced |
|-----------------------------|--|--|-------|----------|
| Security functionality test | Packet filtering | Block specific packets; permit specific packets | ■ | ■ |
| | Traffic flow statistics | Log the flow content, including time, throughput, protocol, and port | ■ | ■ |
| | Security events records | Generate events of violation identified by time, source IP, destination IP, and type | ■ | ■ |
| | Secure administrative interface | Provide password administration; if an invalid password is entered over the threshold, GUI will be locked | ■ | ■ |
| | Backup mechanism | The backup device will take over the operation within 10 seconds after the connection or power is removed | | ■ |
| | Security rule control | Configure security rules to manage the network flow; a security rule contains the IP address, protocol, port, and time | | ■ |
| Self-protection test | Prevention of DoS attacks | The DUT shouldn't hang or reset when under attack; security functions should continue to work after attacks are terminated | ■ | ■ |
| | Recovery from abnormal system shutdown | If the DUT is shut down abnormally, it should work properly after it's reset | ■ | ■ |

Table 7. Evaluation criteria of security functionality and self-protection tests—IDPs.

| Class | Test case | Criteria | Basic | Advanced |
|-----------------------------|--|--|-------|----------|
| Security functionality test | FN/FP test | For abnormal flows, the FN must be less than or equal to 10 percent; for benign flows, the FP must be equal to 0 percent | ■ | |
| | | For abnormal flows, the FN must be less than or equal to 7 percent; for benign flows, the FP must be equal to 0 percent | | ■ |
| | Evasion attacks | The DUT must be able to detect evasion attacks | ■ | ■ |
| | Security events records | Generate events of violation identified by time, source IP, destination IP, and type | ■ | ■ |
| | Secure administrative interface | Provide password administration; if an invalid password is entered over the threshold, GUI will be locked | ■ | ■ |
| | Online updates | The DUT can be updated via the Internet | | ■ |
| | IPv6 packet inspection | The DUT can block IPv6 attacks | | ■ |
| Self-protection test | Prevention of DoS attacks | The DUT shouldn't hang or reset when under attack; security functions should work after attacks are terminated | ■ | ■ |
| | Remote management for abnormal flows | Network services running on the DUT should resist malicious network flows | | ■ |
| | Recovery from abnormal system shutdown | If the DUT is shut down abnormally, it should work properly after it's reset | ■ | ■ |

corresponding documents. The evaluator uses these three tables as guidelines to review DUT-related documents. In the testing phase, DUTs must be evaluated by the security functionality, stress, self-protection, and stability tests' criteria.

Common Defects Found in Document Review

Two ICT product developers provided documentation for the document review's pilot run. We found four defects:

- Neither developer was familiar with the definition of the TOE security function interface (TSFI). TSFI contains the physical and logical interfaces to call security functions for the TOE.
- One developer didn't describe the TOE SFRs completely.
- One developer provided the protection mechanism only for the specific sensitive data, which didn't correspond to the TOE's SFRs.
- The error messages displayed by these two products

Table 8. Test results of firewall security criteria.

| Test case | No. of testing devices | No. of passes | Device A | Device B | Device C | Device D | Device E |
|---------------------------------|------------------------|---------------|----------|----------|----------|----------|----------|
| Packet filtering | 4 | 4 | Pass | Pass | Pass | Pass | N/A |
| Traffic flow statistics | 3 | 3 | N/A | Pass | Pass | Pass | N/A |
| Security events records | 2 | 1 | N/A | Fail | Pass | N/A | N/A |
| Secure administrative interface | 4 | 4 | Pass | Pass | Pass | Pass | N/A |
| Backup mechanism | 3 | 3 | N/A | Pass | Pass | Pass | N/A |
| Security rule control | 3 | 3 | Pass | Pass | Pass | N/A | N/A |
| Prevention of DoS attacks | 2 | 2 | Pass | Pass | N/A | N/A | N/A |
| Throughput test | 2 | 2 | Pass | N/A | N/A | N/A | Pass |
| Maximum connection rate test | 2 | 2 | Pass | N/A | N/A | N/A | Pass |

Table 9. Test results of IDP security criteria.

| Test case | No. of testing devices | No. of passes | Device A | Device B |
|--------------------------------------|------------------------|---------------|----------|----------|
| FN/FP test | 2 | 0 | Fail | Fail |
| Evasion attacks | 2 | 1 | Pass | Fail |
| Security events records | 1 | 1 | N/A | Pass |
| Secure administrative interface | 2 | 2 | Pass | Pass |
| Online updates | 2 | 0 | Fail | Fail |
| IPv6 packet inspection | 2 | 2 | Pass | Pass |
| Prevention of DoS attacks | 2 | 2 | Pass | Pass |
| Remote management for abnormal flows | 2 | 2 | Pass | Pass |
| RealFlow test | 2 | 0 | Fail | Fail |

weren't illustrative enough to show which security functions had failed.

These defects can affect developers' and network administrators' interpretation of configurations and error messages, which could impact security. To overcome these drawbacks, we offer tips for preparing complete documents for review. First, developers should understand which TSFIs are provided by the DUT and explain TSFI messages. They should understand the SFR objectives and fill in the related items in the respective table. Developers should consider situations in which attacks might occur and describe practical methods to prevent those attacks. Finally, they should provide a manual regarding error messages for evaluators.

Common Defects Found in Testing

We used five firewalls and two IDPs in our pilot run of test cases. Tables 6 and 7 illustrate most of those test cases and their evaluation criteria. Both throughput and maximum connection rate tests belong to stress tests. These two test cases check whether the DUT security functionalities work properly when the background traffic achieves the maximum throughput and maximum connection rate, respectively. RealFlow uses replay testing using the same packet traces captured monthly to evaluate DUT stability.

Tables 8 and 9 illustrate the test results—pass or fail—for firewalls and IDPs. Some test cases weren't performed for the specific device during the period of pilot runs; we use "N/A" to denote this situation. Most

test cases have a high pass ratio, but some have a very low ratio. Table 8 shows most firewalls pass all test cases except security event record.

Table 9 shows that all DUTs fail the FN/FP test and online update test cases. We used 109 intrusions, which were published in one year, as samples for the FN/FP test. The test results show that the FN rate for devices A and B are approximately 55 and 98 percent, respectively, meaning neither DUT was able to detect most of the recent attacks. To enhance the intrusion signatures, developers had to extract the new pattern from the attack samples, which caused the failed test. In fact, device A's FN rate had to decrease to less than 10 percent after several rounds of improvement to pass this test case. The reason DUTs fail in the online update test case is that most IDPs don't provide complete and flexible update mechanisms. To pass this test, we suggest that developers refer to the evaluation criteria to modify the scale for setting the online update parameters in DUT firmware.

Zero Initial Pass Ratio in RealFlow

No DUTs passed RealFlow in the initial try, as Table 9 shows. This means that real traffic can easily trigger hidden defects. Developers must fix the defects in several rounds to sustain DUTs over one or two weeks. RealFlow can ensure DUT stability when DUTs pass the test.

Full-spectrum analysis is necessary for security evaluation of ICT products. Compared to existing schemes, the proposed lightweight total security criteria can cover all key test methodologies in terms of security functionality, stress, and self-protection. Instead of artificial traffic, we utilize real traffic captured from an operational network to test the stability of the DUT.

The proposed set of security criteria is useful in the

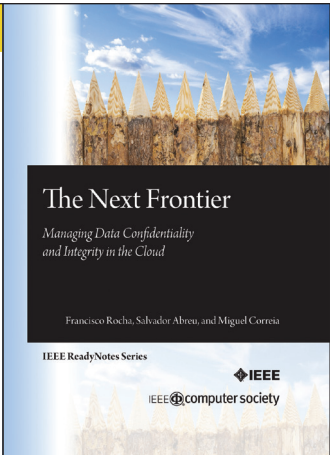
current environment where there are many security products to choose from and rapid but thorough evaluations are necessary. While many government agencies are defining their own national security criteria, our work serves as a reference design for them. For the academic society, the criteria also showcase how to put principles into practices and could draw another stream of research efforts in securing a system in a more comprehensive way. ■


Acknowledgments

Taiwan's National Communications Commission and National Science Council partially supported this work. We thank the editor and the anonymous reviewers for their valuable comments on this article. We also thank our colleagues, I-Wei Chen, Rong-Mao Jheng, Chun-Nan Lu, Fu-Yu Wang, Eric Rui-Bin Wu, and Rex Chin-Lung Wu at Network Benchmarking Lab for their contributions in defining and implementing the security criteria.

References

1. "Security Testing Guide for Information and Communication Devices," Nat'l Communications Commission, Mar. 2012; www.ncc.gov.tw/chinese/law.aspx?site_content_sn=260&is_history=0.
2. "ISO/IEC 15408-Common Criteria," ISO Security Solutions; www.isosecuritysolutions.com/ISOIEC-15408.html.
3. J. Kallberg, "The Common Criteria Meets Realpolitik: Trust, Alliances, and Potential Betrayal," *IEEE Security & Privacy*, vol. 10, no. 4, 2012, pp. 50–53.
4. M.A.M. Isa et al., "Finest Authorizing Member of Common Criteria Certification," *Proc. Int'l Conf. Cyber Security, Cyber Warfare and Digital Forensic (CyberSec 12)*, IEEE, 2012, pp. 155–160.
5. I.W. Chen et al., "Extracting Attack Sessions from Real Traffic with Intrusion Prevention Systems," *Proc. IEEE Int'l Conf. Communications (ICC 09)*, IEEE, 2009, pp. 889–893.
6. Y.D. Lin et al., "On Campus Beta Site: Architecture,



NEW from  **CSPress**

THE NEXT FRONTIER
Managing Data Confidentiality and Integrity in the Cloud
by Francisco Rocha, Salvador Abreu, and Miguel Correia

CS authors present the architecture, main mechanisms, and challenges of their proposed defense against malicious insiders in the cloud.
ISBN 978-0-7695-4978-1 • 7" x 10" • 58 pp.

Order .PDF (\$15):
<http://bit.ly/12Rk6gP>

Order Paperback (\$19):
<http://bit.ly/166DuZr>

Software

On Computing
podcast

www.computer.org/oncomputing



with
GRADY BOOCH

 IEEE 

Designs, Operational Experience, and Top Product Defects,” *IEEE Comm.*, vol. 48, no. 12, 2010, pp. 83–91.

7. R. Prieto-Diaz, *The Common Criteria Evaluation Process: Process Explanation, Shortcomings, and Research Opportunities*, tech. report CISC-TR-2002-003, Commonwealth Information Security Center, James Madison Univ., Dec. 2002.
8. “Network IPS Enterprise Certification Testing Criteria Version 1.4,” ICSA Labs, Nov. 2011; <https://www.icsa-labs.com/icsa/nips>.
9. “Network Intrusion Prevention Systems Test Methodology V6.2,” NSS Labs, 2012; <https://www.nsslabs.com/reports/network-intrusion-prevention-systems-test-methodology-v62>.
10. Y.D. Lin et al., “PCAP Lib: A Framework of Extracted, Classified, and Anonymized Packet Traces,” to appear in *IEEE Systems J*, 2014.

Ying-Dar Lin is a professor in the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan, as well as founder and director of the Network Benchmarking Lab and the Embedded Benchmarking Lab. His research interests include design, analysis, implementation, and benchmarking of network protocols and algorithms; quality of service; network security; deep-packet inspection; P2P networking; and embedded hardware/software codesign. Lin received a PhD in computer science from the University of California, Los Angeles. He’s an IEEE Fellow and on the editorial boards of *IEEE Transactions on Computers*, *Computer*, *IEEE Wireless Communications*, *IEEE Network*, *IEEE Communications Magazine—Network Testing Series*, *IEEE Communications Surveys and Tutorials*, *IEEE Communications Letters*, *Computer Communications*, *Computer Networks*, and *IEICE Transactions on Information and Systems*. Contact him at ydlin@cs.nctu.edu.tw.

Chia-Yin Lee is a postdoctoral researcher in the Information and Communication Technology Labs at National Chiao Tung University. His research interests include cryptography, network security, and image processing. Lee received a PhD in computer science from National Chung Cheng University. He’s a member of IEEE. Contact him at neko.cylee@gmail.com.

Hao-Chuan Tsai is a postdoctoral researcher in the Department of Computer Science at National Chiao Tung University. His research interests include cryptography, image hiding, and malware detection. Tsai received a PhD in computer science from National Chung Cheng University. He’s a member of IEEE. Contact him at saul.tsai@gmail.com.



Executive Committee Members: Dennis Hoffman, President; Phillip Laplante, VP Publications; Alfred Stevens, VP Meetings and Conferences; Marsha Abramo, VP Membership; W. Eric Wong, VP Technical Operations; Joseph Childs, Secretary; Christian Hansen, Treasurer; Jeffrey Voas, Jr., Past President

Administrative Committee Members: Marsha Abramo, Scott Abrams, Loretta Arellano, Lon Chase, Joseph Childs, Pierre Dersin, Irving Engelson, Carole Graas, Lou Gullo, Christian Hansen, Dennis Hoffman, Samuel Keene, Way Kuo, Pradeep Lall, Phil Laplante, Robert Loomis, Rex Sallade, Shihpyng Shieh, Alfred Stevens, Jeffrey Voas, Jr., Pingfeng Wang, and W. Eric Wong

<http://rs.ieee.org>

The IEEE Reliability Society (RS) is a technical Society within the IEEE, which is the world’s leading professional association for the advancement of technology. The RS is engaged in the engineering disciplines of hardware, software, and human factors. Its focus on the broad aspects of reliability, allows the RS to be seen as the IEEE Specialty Engineering organization. The IEEE Reliability Society is concerned with attaining and sustaining these design attributes throughout the total life cycle. The Reliability Society has the management, resources, and administrative and technical structures to develop and to provide technical information via publications, training, conferences, and technical library (IEEE Xplore) data to its members and the Specialty Engineering community. The IEEE Reliability Society has 23 chapters and members in 60 countries worldwide.

The Reliability Society is the IEEE professional society for Reliability Engineering, along with other Specialty Engineering disciplines. These disciplines are design engineering fields that apply scientific knowledge so that their specific attributes are designed into the system / product / device / process to assure that it will perform its intended function for the required duration within a given environment, including the ability to test and support it throughout its total life cycle. This is accomplished concurrently with other design disciplines by contributing to the planning and selection of the system architecture, design implementation, materials, processes, and components; followed by verifying the selections made by thorough analysis and test and then sustainment.

Visit the IEEE Reliability Society Web site as it is the gateway to the many resources that the RS makes available to its members and others interested in the broad aspects of Reliability and Specialty Engineering.

