Reliability

August/September/October 2015

IoT Critical Infrastructure and Computer Security Education

IEEE

ReliabilitySociety

Reliability Society Administrative Committee (AdCom) Members

Officers (Excom)

President
Christian Hansen

Sr. Past President
Jeffrey Voas

Jr. Past President
Dennis Hoffman

Vice Presidents

Technical Activities
Shiuhpyng Winston Shieh

Publications
W. Eric Wong

Meetings and Conferences
Alfred Stevens

Membership
Marsha Abramo

Secretary
Scott Abrams

Treasurer
Bob Loomis

Elected Members-at-Large (with vote)

TERM EXPIRES 2015 (DEC 31)		***TERM EXPIRES 2016 (DEC 31)***		***TERM EXPIRES 2017 (DEC 31)***	
Marsha Abramo	Shiuhpyng Winston Shieh	Lou Gullo	Zhaojun (Steven) Li	Joseph A. Childs	Carole Graas
Loretta Arellano	Alfred Stevens	Christian Hansen	Bob Loomis	Pierre Dersin	Samuel J. Keene
Lon Chase	Rex Sallade	Pradeep Lall	Pradeep Ramuhalli	Lance Fiondella	W. Eric Wong

Standing Committees and Activities/Initiatives

Web Master
Lon Chase

Standards
Lou Gullo

Chapters Coordinator
Loretta Arellano

Professional Development
Marsha Abramo

Constitution and Bylaws
Dennis Hoffman

Fellows
Sam Keene

Finance
Christian Hansen

Academic Education/Scholarship
Sam Keene

Meetings Organization
Alfred Stevens

Membership Development
Marsha Abramo

Transactions Editor
Way Kuo

Newsletter Editor
Joe Childs

Video Tutorials
Christian Hansen

Nominations and Awards
Jeffrey Voas

Life Science Initiative
Peter Ghavami

Transportation Electrification
Sam Keene
Pradeep Lall
Michael Austin

IEEE Press Liaison
Dev Raheja

IEEE Government Relations Committees

Energy Policy
Jeff Voas

Transportation and Aerospace Policy
Scott Abrams

Medical Technology Policy
Jeff Voas

Critical Infrastructure Protection
Sam Keene

Career and Workforce Policy
Christian Hansen
(corresponding only)

Intellectual Property Committee
Carole Graas
(corresponding only)

Research and Development Committee
Pradeep Lall

Combined TAB Ad Hoc Committee on Attracting Industrial Members
Dennis Hoffman

2013 TAB Awards and Recognition Committee (TABARC)
Dennis Hoffman

Technical Committees

Vice President for Technical Activities

Shiuhpyng Winston Shieh, National Chiao Tung University

Email: ssp@cs.nctu.edu.tw

Technical Committee on Internet of Things (IoT)

Chair: **Jeffrey M. Voas**, National Institute of Standards and Technology

Email: jeff.voas@nist.gov

Co–chair: **Irena Bojanova**, National Institute of Standards and Technology

Email: irena.bojanova@nist.gov

Committee Member:

1. George F. Hurlburt: CEO of Change Index

Technical Committee on System and Software Assurance

Chair: **Eric Wong**, University of Texas at Dallas

Email: ewong@utdallas.edu

Technical Committee on Prognostics and Health Management (PHM)

Chair: **Rex Sallade**, Sikorsky Aircraft Co.

Email: Rex.Sallade@SIKORSKY.COM

Co–chair: **Pradeep Lall**, Auburn University

Email: lall@eng.auburn.edu

Technical Committee on Big Data

Chair: **David Belanger**, Stevens Institute of Technology

Email: david.belanger@stevens.edu

Technical Committee on Trustworthy Computing and Cybersecurity

Chair: **Wen-Guey Tzeng**, National Chiao Tung University

Email: wgtzeng@cs.nctu.edu.tw

Committee Member:

1. Raul Santelices: Assistant Professor, Department of Computer Science and Engineering, University of Notre Dam
2. Brahim Hamid: Associate Professor, IRIT Research Laboratory, University of Toulouse, France

Technical Committee on Reliability Science for Advanced Materials & Devices

Chair: **Carole Graas**, Colorado School of Mine & IBM Systems and Technology Group

Email: cdgraas@mines.edu

Technical Committee on Systems of Systems

Chair: **Pierre Dersin**, Alstom Transport

Email: pierre.dersin@transport.alstom.com

Technical Committee on Resilient Cyber-Systems

Chair: **Pradeep Ramuhalli**, Pacific Northwest National Laboratory

Email: pradeep.ramuhalli@pnnl.gov

Standards Committee

Chair: **Louis J Gullo**, IEEE RS Standards

Email: Lou.Gullo@RAYTHEON.COM

Committee Member:

1. Ann Marie Neufelder: Softrel, LLC – Owner; IEEE P1633 Standard Working Group Chair
2. Lance Fiondella: Assistant Professor, Dept. of Electrical and Computer Engineering, University of Massachusetts Dartmouth; IEEE P1633 Standard Working Group Vice Chair
3. Steven Li: Assistant Professor, Industrial Engineering and Engineering Management, Western New England University; IEEE P61014 Standard Working Group Chair
4. Diganta Das: Research Staff at the Center for Advanced Life Cycle Engineering, University of Maryland
5. Sony Mathews: Engineer at Halliburton, IEEE P1856 Standard Working Group Chair
6. Mike Pecht: Chair Professor and Director of Center for Advanced Life Cycle Engineering, University of Maryland
7. Arvind Sai Sarathi Vasan: Research Assistant at Center for Advanced Life Cycle Engineering , University of Maryland; IEEE P1856 Standard Working Group Vice-Chair
8. Joe Childs: Staff Reliability/Testability Engineer at Lockheed Martin

Working Group on Education

Chair: **Zhaojun (Steven) Li**, Western New England University

Email: zhaojun.li@wne.edu

Committee Member:

1. Emmanuel Gonzalez, Jardine Schindler Elevator Corporation

Editorial Board

Editorial Staff

Message from the Editor

With the fast growth of IoT technology, new reliability concerns also emerged. Among the new concerns, we will study two particularly interesting topics in this issue including IoT critical infrastructure and computer security education.

On the first topic, P. A. Laplante of Penn State University presents an article *"Critical Infrastructure Systems and the Internet of Things"* to discuss the considerations for IoT applications that involve or interact with critical systems. In the forthcoming IoT age, traditional security concerns have been extended to threaten new kinds of devices and systems. Fu-Hau Hsu introduces in his article *"Is Your Vehicle Controlled by You?"* that the new threat can be physically harmful when a vehicle is hijacked given that hacking is done against vehicular computers. Wireless networks are part of the critical infrastructure in the IoT age. To illustrate an applicable testbed for security analysis, Borting Chen and Yu-Lun Huang's article *"Launching a Security Testbed for Wireless Networks with Extensibility to Support Mobile Experiments"* presents relevant concepts, case studies, performance measurement, and war-driving experiments.

Computer security education has been a hot topic for decades. Foreseeing the trend, computer security becomes even more important in the IoT age since the large-quantity and the heterogeneity of devices make security and reliability problems more and more complex. An article "*Tool, Technique, and Tao in Computer Security Education.*" provided by Zhenkai Liang and Jian Mao discusses fundamental concepts and the right methodology for computer security education. It points out a good direction for the students who are facing the fast-changing security landscape. "*CTF: Alternative Training for Offensive Security*", the last article in this issue written by Chung-Kuan Chen and myself, describes an alternative way for computer security education. Traditional security education is insufficient for offensive security. Most security courses are designed from the defender's perspective, and have a big gap from the real world problems in practice. The emerging contest Catch-The Flag (CTF) can complement the traditional education for security training. In a CTF contest, competitors will try to solve security problems that are close to real world problems, or attack competitors' systems with a wide range of knowledge of skills.

I sincerely invite you to read the interesting articles in this issue, and provide us with your comments and feedback. Happy reading.

Shiuhpyng Winston Shieh
Editor-in-Chief, IEEE Reliability

Reliability

IEEE Reliability Society

Regular Issue

Critical Infrastructure Systems and the Internet of Things

Phillip A. Laplante
Pennsylvania State University
plaplante@psu.edu

Abstract — **Over the last few years there have been many publications in the technical literature and popular press heralding spectacular applications of the Internet of Things (IoT). In fact, the IEEE has several publications and a major strategic initiative involving the IoT and the Reliability Society has been actively involved in this work [1]. The US National Institute of Standards and Technology (NIST) also has a significant initiative on IoT, and I have been participating in this work.**
Based on my own work with NIST and my work in licensing of software engineers in the US e.g. [2], I want to discuss some considerations for IoT applications that involve or interact with critical systems.

Keywords — **internet of things, IoT, critical systems**

I. Introduction

IoT systems contain specialized sensors, computing devices and actuators operating in some environment. Jeff Voas has proposed a more formal model for an IoT based on a set of primitives [3]:

1. sensor -- an electronic utility that digitally measures physical properties such as temperature, acceleration, weight, sound (e.g. cameras and microphones),
2. snapshot -- an instant in time,
3. cluster -- a grouping of sensors that can appear and disappear instantaneously,
4. aggregator – a software implementation based on mathematical function(s) that transforms sensor data into intermediate data,
5. weight – the degree to which a particular sensor's data will impact an aggregator's computation,
6. communication channel – any medium by which data is transmitted (e.g. wireless or wired),
7. external utility (eUtility) – a software or hardware product or service which executes processes or feeds data into the overall dataflow of the NoT,
8. decision trigger -- creates the final result or results from data concentrations and any other data needed to satisfy the purpose and requirements of a specific NoT [3].

This formal framework is already proving to be quite useful in characterizing systems, helping identify gaps during requirements elicitation, proving properties of the system, and helping engineers to design test cases, among many other uses.

Although relatively new, there are many deployed instances of IoTs. Typical applications include smart homes, smart cities, automobiles, and critical infrastructure systems such as water treatment, power generation and distribution. Many of these systems are experimental and small in scale as platform builders and end users are still learning about the challenges of building real IoT systems. From my discussions with implementers of real, industrial strength IoT systems, I learned that one of the biggest challenges is the difficulty in meeting the hyped capabilities promised by pundits. For example, achieving desired levels of availability and reliability for even the simplest IoT system is quite challenging. So when we consider and IoT for a critical infrastructure IoT systems we need to be very careful about the gap between theory and reality since the consequences of failed critical systems can be significant.

II. Critical Systems

The notion of a "critical" system is subjective, even if we base our definition on the degree to which a failure affects humans. For example, it is clear that a system failure at a nuclear plant could have mortal consequences. But it is not so clear for some kind of financial system hack that results in significant loss of funds– while no physical injury results, the impact on the lives of those who lost their savings would be significant, even "critical." For the purposes of this discussion, let's consider critical systems to include:

- telecommunication infrastructure,
- water supply,
- electrical power system,
- oil and gas,
- road transportation,
- railway transportation,
- air transportation
- banking and financial services,
- public safety services,
- healthcare system,
- administration and public services.

This list is not arbitrary – it's based on a US Congressional Report [4]. My own research interests include IoT systems in healthcare, for example, to assist patients with Alzheimer's disease.

Designers and builders need to take special precautions to ensure the safety of critical systems. But what about non-critical systems that might inadvertently interact with a critical system in an IoT and cause a catastrophic failure? Even if a certain IoT is not intended for a critical application the potential for unforeseen interaction with a life critical system is possible. In fact the IEEE definition of IoT concedes the notion that all system in an IoT might be connected: "The Internet of Things (IoT) is a computing concept where all things, including every physical object, can be connected - making those objects intelligent, programmable and capable of interacting with humans."

This scenario is very likely. In one demonstration, "hackers" where able to take control of the braking apparatus of a Jeep car, causing it to crash. But the attack vector was through software written to control the audio system [5]. And the Online Trust Alliance (OTA), whose members include Microsoft, Symantec, and Verisign, warn about this kind of problem and have published guidelines in this regard [6]. I would go further and suggest that every IoT system needs to be engineered as if it is a life critical system because of the potential for interaction (either planned or inadvertent) with a critical one.

III. Failure / Fault Database

To help systems designers building critical applications in the IoT we need a national database for IoT incident and vulnerabilities, similar to the National Vulnerability Database hosted by NIST [7]. We could use the NIST primitive set to key the database items. Such a database could:

1. help requirements engineers to anticipate hazards, write misuse and abuse cases, and create antipatterns,
2. assist designers in the IoT create more fault tolerant systems
3. facilitate creation of sensor fusion algorithms and techniques for dealing with uncertain data,
4. enable prognostic health management in sensor networks,
5. guide test engineers in developing test cases,
6. help users of the IoT understand the limitations of real systems.

And there would many other uses.

As a step in this direction and as part of my work with NIST, my son and I have created an experimental database of reported sensor failures in deployed IoT applications. Some of the reports are fascinating, for example, sensors being submerged by 100 year flood levels in an environmental IoT. The site is in the alpha testing phase, enters beta testing soon and will be made public by the end of the year.

In the meantime, if you have direct involvement with a deployed IoT and it has experienced some kind of sensor failure – whether due to device physics, environmental factors, mishap, or whatever -- I'd like to include that information in the sensor failure database. If you are interested, please email me and I will send you details. If requested, certain information can be kept confidential.

References

[1] IEEE Internet of Things Community, http://iot.ieee.org/, 2015.

[2] P. Laplante, "When does software affect the health, safety and welfare of the public?," Reliability Society Newsletter, http://rs.ieee.org/images/files/newsletters/2012/1_2012/Phil%20Laplante htm, February, 2012.

[3] J. Voas, "Foundations of the Internet of Things," http://dslsrv.gmu.edu/The%20Foundations%20of%20IoT_v2.5.pdf, 2015.

[4] J. D. Moteff and P. Parfomak, "Critical Infrastructure and Key Assets: Definition and Identification," Congressional Research Service, Library of Congress, 2004.

[5] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway -- with Me in It," Wired, http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, July 21, 2015.

[6] R. Chirgwin, "IoT Security is Rubbish Says IoT Vendor Collective," The A Register, August 12, 2015 http://www.theregister.co.uk/2015/08/12/iot_security_is_rubbish_says_iot_vendor_collective/, 2015

[7] National Vulnerability Database, https://nvd nist.gov/, 2015.

Phillip A. Laplante is Professor of Software Engineering at The Pennsylvania State University. He received his B.S., M.Eng., and Ph.D. from Stevens Institute of Technology and an MBA from the University of Colorado. He is a Fellow of the IEEE and SPIE and has won international awards for his teaching, research and service. Since 2010 he has led the effort to develop a national licensing exam for software engineers.

He has worked in avionics, CAD, and software testing systems and he has published 27 books and more than 200 scholarly papers. He is a licensed professional engineer in the Commonwealth of Pennsylvania and a Certified Software Development Professional. He is also a frequent technology advisor to senior executives, investors, entrepreneurs and attorneys.

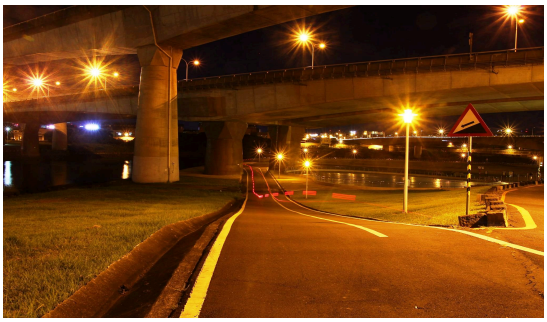
His research interests are in software testing, requirements engineering and software quality and management.

Prior to his appointment at Penn State he was a software development professional, technology executive, college president and entrepreneur.

More information can be found at www.personal.psu.edu/pal11

Is Your Vehicle Controlled by You?

Fu-Hau Hsu
National Central University
hsufh@csie.ncu.edu.tw



***Abstract* — To increase the vehicle security and the comfort of drivers and passengers, many electronic devices, embedded systems, and vehicle computers have been added to automobiles to control the automobiles and various devices. However, ironically, research shows that attackers can take over the control of a moving vehicle from its driver through wireless networks by compromising these electronic devices or vehicle computers. Even though the anti-virus industry and security researchers have developed many approaches, such as anti-virus software and patches, to defend a system against malware and computer attacks, these approaches are not suitable for vehicle computer security, because they are not able to provide real-time defense and achieve zero-tolerance upon false positives and false negatives. In this new battlefield, for security researchers and security industry, there are plenty of tough battles ahead of them.**

***Keywords* — Vehicle Security, Vehicle Computer, Car Hijacking.**

When security industry is doing their best to provide a secure and reliable environment for desktop, laptop, and smartphone users, a much more vital security problem, *vehicle computer security*, is emerging, which may not only cause life-threating problems but also become a powerful tool for terrorists or criminals.

In a demonstration [1] made in 2015, Charlie Miller and Chris Valasek showed that through Internet and wireless connections, they can easily remotely control a Jeep Cherokee driven by a human user. In other words, just using a laptop, they remotely disabled a driver's capability to control his Jeep and got full control of the dashboard functions, steering, brakes, and transmission of that car. Besides, they also obtained the Jeep's GPS coordinates, which leaked the positions of the Jeep. This demonstration proves that Checkoway *et al.* [2]'s 2011 work is not paranoid speculation and can be further enhanced.

In the above demonstration, Miller and Valasek utilized the vulnerability in an electronic control unit (ECU) of a Jeep to rewrite the firmware of a chip with their own code. Based on the instructions issued by them, the code used the CAN bus [3, 4] to send commands to the physical components, such as the engine and wheels, to hijack the Jeep. As vehicles incorporate more electronics to provide more functions to their users, more vulnerabilities are also unwittingly introduced to vehicles. As a result, nowadays, car thieves can utilize a completely different approach [5] to steal cars, especially keyless cars. For example, a car thief can use a device to physically access a car's OBD (on-board diagnostics) connector to collect unsecured key code. With the key code, the thief can quickly create a new car key and use the new key to star the car engine and drive it away.

Even though the above situations have inspired lawmakers' intention to make related bills and set new digital security standards for cars and trucks [6], the fundamental solution still should come from technology. However, due to having different hardware, software, and communication protocols in an automobile and zero tolerance upon false negatives, the traditional antivirus models or patch mechanisms may not be suitable for vehicle computers or vehicle embedded systems. First of all, in a traditional antivirus model, it may take a couple of weeks to obtain a signature of a zero-day malicious program. And it may take several months to find a vulnerability of a program. Besides, extra time is required to obtain related patch. However, this time period is too long for vehicles. After all, an unfixed vehicle is vulnerable to car hijack. A hijacked car may result in life-threating problems and cause huge panic upon the public. Moreover, even if a signature or a patch is created, it is not easy to dispatch them to all related antivirus systems or software on automobiles. Automobile manufacturers usually recall faulty cars and ask their technicians to use special devices to fix problems. However, the recall-and-fix pattern means that some cars may remain unfixed because their owners may ignore or do not know a recall. The patterns also means that it needs to take a much longer time to fix software vulnerability or add a malware signature to an antivirus system in a vehicle. As a result, if we do not develop new approaches

to handle vehicle computer security problems, it is very likely that cars with unpatched software or cares with un-updated antivirus software may be full of the streets, which creates a serious security problem.

We believe that Checkoway *et al.*, Miller, and Valasek just unveiled a new type of critical security threats that will last for a long period of time. In this new battlefield, for security researchers and security industry, there are plenty of tough battles ahead of them.

REFERENCES

[1] Andy Greenberg, "Hackers Remotely Kill a Jeep on the Highway—With Me in It," http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[2] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," USENIX Security, San Francisco, CA, USA, August 10–12, 2011.

[3] Steve Corrigan, "Introduction to the Controller Area Network (CAN)," http://www.ti.com/lit/an/sloa101a/sloa101a.pdf

[4] AA1Car, "How to Diagnose a Controller Area Network (CAN)," http://www.aa1car.com/library/can_systems.htm

[5] Bill Howard, "Hack the diagnostics connector, steal yourself a BMW in 3 minutes," http://www.extremetech.com/extreme/132526-hack-the-diagnostics-connector-steal-yourself-a-bmw-in-3-minutes

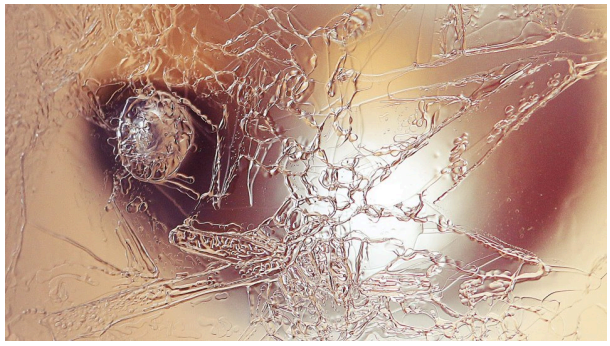
[6] Dan Goodin, "Senator: Car hacks that control steering or steal driver data way too easy," http://arstechnica.com/security/2015/02/senator-car-hacks-that-control-steering-or-steal-driver-data-way-too-easy/

Fu-Hau Hsu received his Ph.D. degree in Computer Science from Stony Brook University, New York, USA in 2004. He is an associate professor in the Department of Computer Science and Information Engineering at National Central University, Taiwan, R.O.C. His research interests include system security, smartphone security, web security, information hiding, operating system, and networking.

Launching a Security Testbed for Wireless Networks with Extensibility to Support Mobile Experiments

Borting Chen
National Chiao Tung University
bortingchen.ece98g@nctu.edu.tw

Yu-Lun Huang
National Chiao Tung University
ylhuang@cn.nctu.edu.tw



***Abstract*—Creating a wireless testbed for security analysis is more difficult than creating a wired one because it could face stricter challenges in isolating radio signals, guaranteeing fidelity, relocating devices, reproducing or scaling up an experiment. To handle these challenges, we redesign a wireless testbed called Secure Wireless Overlay Observation Network (SWOONv2) for security experiments. SWOONv2 tackles the aforesaid challenges by emulating wireless networks over wired networks. SWOONv2 uses a virtual wireless network interface, which transmit and receives wireless packets over Ethernet cables to guarantee isolation and fidelity during an experiment and to support the scalability requirement. SWOONv2 also adopts a wireless network simulator to simulate the distance between devices and the signal attenuation during transmission, and to support device relocation in order to reproduce a specific experiment. In the paper, two case studies, performance measurement and wardriving experiment, are presented to demonstrate how SWOONv2 supports isolation, fidelity, relocation, repeatability and scalability for wireless security analysis. We also discuss the extensibility of SWOONv2 in an LTE (Long Term Evolution) network for security tests and analysis.**

***Keywords*—Network Testbed, Wireless Security**

I. Introduction

Security analysis is essential when developing network applications, protocols, and systems because the cost of repairing a security flaw is enormous. HP's study [1] shows that fixing defects of software after deployment is twice or more expensive than fixing them during development. Owing to this, researchers generally run various security tests before releasing their development. A test network is hence required to allow researchers to implement their development and conduct test experiments.

In the past few years, various approaches have been presented to create a test network for wireless security analysis, such as network simulation and network emulation. Among these approaches, network simulation is widely accepted by wireless researchers because it can provide an experimental environment without creating a real network. Network emulation which runs simulation in conjunction with real network hardware can create an emulated wireless network in a laboratory environment and provide more fidelity for tests and analysis. Researchers can implement protocols and applications on real network devices to evaluate their development against security requirements due to real network devices are used in a test. Once a test case is tested under practical conditions, researchers can ensure that their development can perform as expected in a real network.

Though network emulation can provide many benefits for security analysis, the available resources of a test network limit the scale of an experiment. To improve the scalability, some researchers federate resources of multiple test networks to form a "testbed" [2], [3], [4]. A testbed can share its resources with fellow researchers and support concurrent experiments. Some testbeds also support flexible configuration like a network simulator, allowing a user to create a test network that has arbitrary topologies to meet different testing requirements.

In general, constructing a testbed should meet several requirements, such as fidelity, isolation, scalability, etc. [5]. When designing a secure wireless testbed, some of the requirements should face a stricter constraint to support security analysis, as listed below.

TABLE I
COMPARISON OF DIFFERENT TESTBEDS

	Emulab	Orbit	MiNT-m	Mobile Emulab	CMU testbed	ASSERT	SWOON
Isolation	X	X	X	X	O	O	O
Fidelity	O	O	O	O	O	O	Δ^2
Relocation	X	X	O	O	Δ^1	Δ^1	O
Repeatability	X	X	X	X	O	O	O
Scalability	O	O	O	O	X	X	Δ^3

1: The distance between wireless devices is simulated by adjusting signal strength.
2: Link-layer emulation is not well supported.
3: Using two machines to emulate a wireless device halves the scalability of SWOON.

- **Isolation**
  Because wireless packets are transmitted over radio waves, a testbed should adopt a stronger isolation mechanism to prevent an experiment from internal pollution (traffic from other experiments) or external interference (traffic from public networks). It is also necessary to block experiment traffic inside an isolated environment and to prevent the malicious packets generated by a test case from being leaked to public networks.
- **Fidelity**
  A wireless testbed may need to provide real hardware to emulate network devices and to support a developer in measuring performance with real hardware constraints. Moreover, some experiments need to capture wireless packets from the link layer to analyze the behavior of wireless transmission. It is necessary for a wireless testbed to faithfully emulate the behavior of a real network interface and allow a user-space application, such as *tcpdump*, to record the packets transmitted in a wireless network.
- **Relocation**
  The spatial distance between source and destination normally determines the reachability of a wireless packet. This requires a testbed to have the ability to change the location of network devices for different testing requirements. Some tests may even require a testbed to relocate the devices during an experiment for realizing the roaming functionality. Therefore, relocation support is indispensable for a wireless testbed.
- **Repeatability**
  Reproducing an experiment requires the support of control over network conditions. However, some wireless network conditions may be affected by environmental factors, for example, signal attenuation is affected by the temperature and humidity of a testing environment. Hence, a testbed needs to work in conjunction with some simulation methods to control wireless network conditions and eliminate the effect of environmental factors.
- **Scalability**
  Scaling up a wireless experiment is difficult because larger space is required to accommodate the testbed. More, a tester should be able to deploy arbitrary numbers of wireless devices in the same subnet, and emulate the change of distances among the wireless devices. The scale of a wireless network hence should not be restricted by the hardware emulating the wireless environment.

In the following sections, we introduce the existing testbeds and compare them in terms of the above requirements.

II. THE EXISTING WIRELESS TESTBEDS

Various approaches have been proposed in previous work to fulfill the above requirements when designing a wireless testbed for security analysis. This section lists several wireless testbeds presented in the past few years:

- **Emulab** [6] deploys wireless devices in a building to allow a user to allocate them for experiments. These wireless devices are immovable and could be interfered by the public wireless network during an experiment.
- **ORBIT** [7] builds a grid of wireless devices and realizes device roaming by transferring the state of a virtual device from one physical device to another. However, no isolation mechanism is adopted between wireless devices makes the testbed hard to partition its resources for concurrent experiments.
- **MiNT-m** [8] and **Mobile Emulab** [9] use robots to provide mobility for wireless devices. However, the space for roaming is restricted by the room size where the testbeds are located.
- **CMU testbed** [10] and **ASSERT** [11] connect the antenna of a wireless device to an FPGA board and leverage signal attenuation to simulate the distance between wireless devices. Although connecting antennas to an FPGA board isolates the test environment from interference, the scale of an experiment is still limited by the number of attenuators that the FPGA board supports.
- **SWOON** [12] uses two wired machines to emulate a wireless device. One machine runs an OS and applications of the device and the other machine emulates a wireless interface for the device. Such a design halves the scalability of a testbed. Besides, some link-layer analysis, such as analyzing wireless headers, cannot be performed on SWOON because the two machines communicate with Ethernet protocol.

Emulab and Orbit install real wireless devices for conducting experiments, which implies the relative distance between devices cannot be adjusted to fulfill the relocation requirement. Emulab and Orbit satisfy the fidelity and scalability requirements, but fail to fulfill isolation and repeatability requirements because the experiment devices are installed in public space without any protection. MiNT-m and Mobile Emulab address the relocation requirement by adopting robots to support mobility, but still suffer from isolation and repeatability problems. CMU testbed and ASSERT leverage FPGA boards to support isolation and repeatability for wireless experiments, but sacrifice the scalability of an experiment. SWOON addresses most of the testbed requirements, but it still has room for improvements in fidelity and scalability. Table I compares the above testbeds in terms of the requirements mentioned in the previous section.

Because SWOON uses wired devices to emulate wireless devices, SWOON can be used to conduct an experiment adopting hybrid networking technologies. The next section introduces a revision of SWOON (SWOONv2) that supports link-layer emulation, reduces the cost in scaling up an experiment, and emulates hybrid networks for more test scenarios. (In the following sections, SWOONv1 is used to represent the research in [12] and SWOONv2 is used to indicate our new design. If only SWOON is presented, it means the description is valid for both versions.)

III. SWOONv2

SWOON is a comprehensive and flexible testbed designed for large-scale wired and wireless experiments. It provides an isolated environment for security experiments and guarantees no interference among the experiments. To support performance measurement with physical hardware constraints, SWOON uses real machines to emulate network devices including wired and wireless devices. A tester can rent machines from SWOON and construct arbitrary network topologies for the machines to meet the test requirements.

Taking DETER [5] as basis, SWOON inherits DETER's infrastructure, management functions, and security mechanisms to provide wired network emulation. In addition, SWOON emulates a wireless network interface (WNIC) on an experiment machine to support wireless network emulation. As described in Section II, SWOONv1 can fulfill most requirements listed in Section I, but it does not support the link-layer emulation and needs double costs in constructing an experiment, which hurts the scalability mentioned in Section I. The following paragraphs explain how we revise SWOONv1 to better fulfill the fidelity and scalability requirements.

- **Wired Network Emulation**
  Similar to SWOONv1, SWOONv2 also emulates a wired network by connecting devices of the same subnet to a VLAN. Packets are generated and routed inside the VLAN to allow a user to monitor real network traffic and conduct performance measurement. The VLAN technology also provides isolation for different experiments and for different subnets in an experiment. This enables a tester to conduct high-risk experiments in SWOONv2, such as analyzing the behavior of malware.

- **Wireless Network Emulation**
  Wireless traffic is transmitted over radio waves, hence, wireless packets can be picked up by any receiver within the radio coverage. This makes a wireless experiment hard to be isolated. To faithfully emulate the broadcast characteristic while ensuring isolation, SWOONv2 installs a Virtual WINC on the experiment machines for emulating a wireless device. The Virtual WINC transmits and receives wireless packets through a dedicated VLAN, which emulates the transmission medium of a wireless network. The dedicated VLAN can emulate different types of network according to the testing requirements. When forwarding a wireless packet to the dedicated VLAN, the Virtual WINC encapsulates the packet with an Ethernet header containing a broadcast IP address. The dedicated VLAN is then treated as a wireless transmission medium, which broadcasts the packet to all Virtual WINCs connecting to the dedicated VLAN. In order to simulate radio coverage, a Wireless Network Simulator (wnSim) is installed inside each Virtual WINC to help determine the reachability of wireless packets. The wnSim allows a user to set up a location for each experiment machine in a virtual space to simulate the distance between wireless devices. When receiving a wireless packet from the dedicated VLAN, the wnSim invoked and the signal attenuation is calculated in terms of the distance between the sender and the receiver. If the receiver is out of sender's transmission range, the wnSim will drop this packet. Such a design helps a user to set up a wireless experiment without any spatial constraint and relocate experiment machines during an experiment.

SWOONv2 leverages the VLAN technology to emulate different types of network and construct network topologies required in an experiment. The VLAN technology creates a Faraday cage-like environment, shields the emulated wireless network from signal interference, and prevents wireless traffic from being leaked to public networks. Emulating wireless networks over VLANs also facilitates SWOONv2 to support scalability. A user who wants to create a larger experiment can simply connect more wireless devices to the dedicated VLAN without considering hardware constraints. Additionally, the emulated WNIC mentioned above acts as a real WNIC in processing wireless packets, which can support the link-layer emulation and improve the fidelity of SWOONv2. Only one machine is required to emulate a wireless device also mitigates the emulation cost presented in SWOONv1 and improves the scalability.

IV. Applications

In the following paragraphs, two security analysis conducted on SWOONv2 are presented. The first case describes a performance measurement test, and we also depict how an experiment is set up on SWOONv2 in this case study. In the second case, a wardriving experiment is presented, and we adopt a network sniffing tool to monitor wireless transmission and analyze the received wireless packets.

A. Case I: Performance Measurement

In the first case, an emulated network is created to measure the performance of an 802.1X authentication protocol, One-time key Secure Network Protocol (OSNP) [13]. This experiment demonstrates that SWOONv2 can fulfill the fidelity requirement in measuring performance.

· **Setup**

Creating an experiment on SWOONv2 involves three steps: (1) design a network topology used in a real-world experiment, (2) map the real-world topology to a SWOON experiment topology, and then (3) describe the experiment topology in a Tcl/Tk script. To create an experiment for testing OSNP, we first design a real-world topology, which contains the following network devices.

- KDC1 and KDC2 are key distribution centers (KDCs) for authentication.
- SS1 and SS2 are service servers.
- Switch1 and Switch2 connect devices of LAN and WAN respectively.

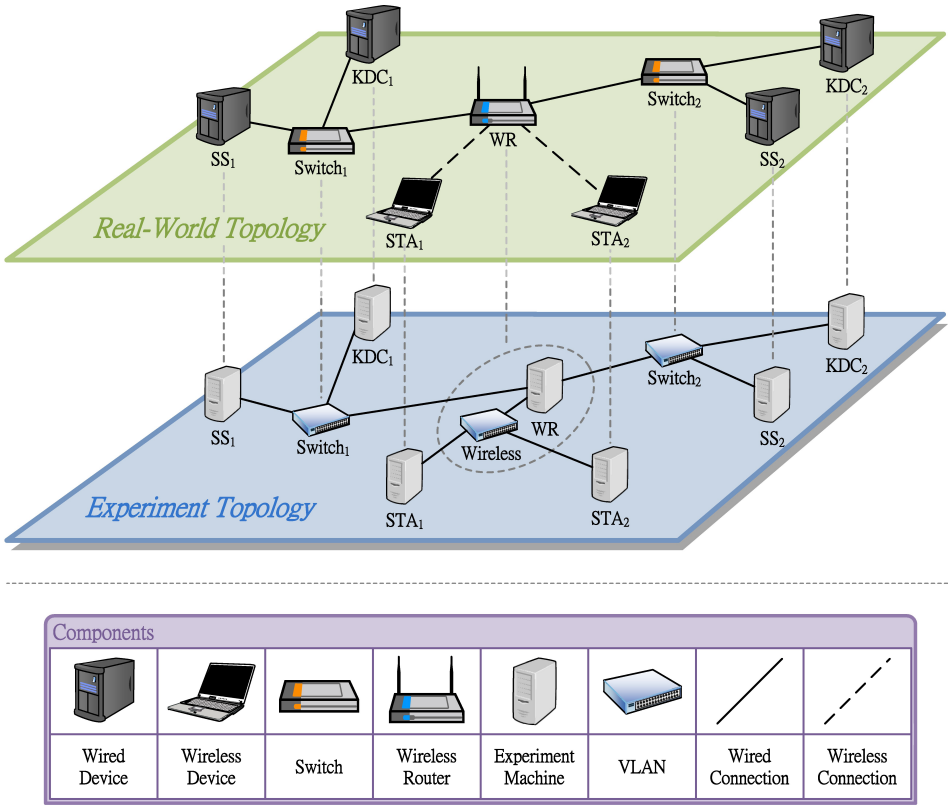


Fig. 1. The topologies for conducting an OSNP experiment: (upper) in real world & (lower) on SWOONv2.

TABLE II
THE NUMBER OF EXCHANGED MESSAGES AND TIME COST OF AUTHENTICATION PROTOCOLS [13]

	OSNP			TLS	TTLS/ MD5	PEAPv0
	Initial	Subsequent	Handover			
Messages	4	4	4	14	12	18
$\Delta t(ms)$	10.37	4.43	10.68	69.47	59.32	66.24

- WR is a wireless router having two extra NICs connecting to LAN and WAN.
- STA1 and STA2 are wireless stations attempting to associate with the wireless router.

The connections between these network devices are shown in the "Real-World Topology" of Fig. 1 (the upper part). Next, the real-world topology is mapped to a SWOON experiment topology, as the "Experiment Topology" shown in Fig. 1 (the lower part). Each network device in the "Real-World Topology" (except for the switches) is emulated by one experiment machine. Hence, seven experiment machines are required to emulate a wireless router, two wireless stations, two service servers, and two KDC servers. Two VLANs (managed by Switch1 and Switch2) are used to emulate switches and are connected to the wireless router and other wired devices, as the connections shown in the "Real-World Topology." Finally, an additional VLAN (Wireless) is created to serve as a dedicated VLAN and emulate a wireless network. The wireless router and the wireless stations are connected to the dedicated VLAN for communicating wirelessly.

Based on the "Experiment Topology," the number of required experiment machines and the connections between VLANs and experiment machines can be determined. The "Experiment Topology" is then described using the Tcl/Tk language and the information is saved in a script file. After submitting the file to SWOONv2, a server al-locates experiment machine and configures VLANs on the Experiment Switch. The required experiment network is hence created.

· **Results**

To measure the time cost of the authentication process, the *wpa_supplicant* [14] is installed on the wireless stations and is patched to support OSNP. A daemon is also implemented in the wireless router and the KDC servers to provide OSNP authentication. During the experiment, the wireless stations try to associate with the router and the time cost of the authentication process is measured. Tables II lists the number of exchanged messages and the time cost of OSNP. We also compare the results with the performance of other authentication protocols. (The complete results are described in our previous publication [13].) The time cost of authentication is determined by 1) the number of exchanged messages, 2) the processing time on the servers and 3) the transmission delay between experiment machines. Note that, each network device in a SWOON experiment is running by a real machine and real traffic is transmitted among the experiment machines. This shows that the time cost measured in a SWOON experiment is convincing and SWOONv2 is a

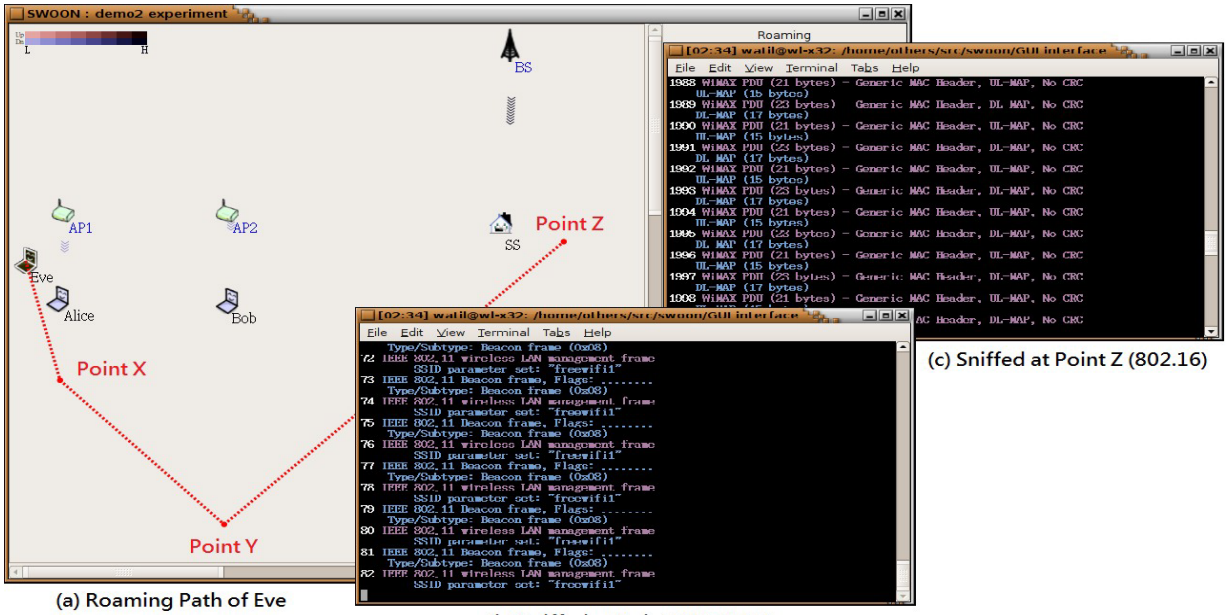


Fig. 2. Sniffing wireless traffic in a wardriving experiment.

favorable choice for measuring the performance of a wireless protocol.

B. Case II: Wardriving Experiment

In the second case, a wardriving experiment is conducted in a wireless network supporting 802.11 and 802.16 protocols to demonstrate that SWOONv2 supports device relocation and isolation.

- **Setup**

To create a wardriving experiment, we allocate seven machines to emulate two access points (AP1 and AP2), three wireless stations (Alice, Bob, and Eve), one WiMAX base station (BS) and one WiMAX subscriber station (SS). The steps of experiment creation are similar to those of the first case study, except that two dedicated VLANs are allocated: the first for Wi-Fi transmission and the second for WiMAX transmission. Alice, Bob and Eve are equipped with one Virtual WNIC for Wi-Fi communication, and Eve is installed with an additional Virtual WNIC supporting 802.16 protocol. Signal strength and device locations are configured in the wnSim to simulate the signal attenuation during transmission.

- **Results**

We use the GUI provided by SWOONv2 to track the wire-less stations and configure their locations during the experiment. Fig. 2 shows the GUI and the screenshot captured during the experiment. Eve, the eavesdropper, executes *tcpdump* and roams in the virtual space along the dotted line shown in Fig. 2 (a). When reaching point X, Eve can sniff the Wi-Fi sessions sent among Alice, Bob, AP1 and AP2. No WiMAX packet can be received because Eve is out of the transmission range of WiMAX stations. When reaching point Z, only the WiMAX traffic sent between the BS and the SS can be received, and Eve cannot hear the communication between Alice and Bob any more.

The effect of location change is simulated by wnSim. When Eve roams in the virtual space, the wnSim recalculates the virtual distances between Eve and other wireless devices. This causes Eve's wnSim drops Wi-Fi packets when it roams out of Alice's and Bob's transmission ranges. Such an experiment shows that a user can leverage wnSim to deploy devices in an emulated wireless network without any spatial constraint and relocate wireless devices during an experiment.

The sniffing results of Wi-Fi and WiMAX networks are presented in Fig. 2 (b) and (c) respectively. These figures show that 802.11 and 80.216 MAC headers are attached when packets are transmitted in the wVLANs. It also demonstrates that Eve can use *tcpdump* to capture and parse these link-layer frames adequately. This enables a user to run various wireless attacks on SWOONv2, ranging from application layer (e.g. man-in-the-middle attack) to link layer (e.g. de-authentication attack), and use a sniffing tool to analyze the behavior of the wireless attacks.

C. Analysis

We compare SWOONv2 with other wireless testbeds in terms of supporting the two security experiments mentioned above. The comparison results are listed in Table III. All testbeds can support high-fidelity performance measurement in the first case study because real network hardware is used to emulate hardware constraint of a wireless device. However, the results of Emulab, ORBIT, MiNT, and Mobile Emulab may have little deviation because no isolation mechanisms are adopted in these testbeds to prevent an experiment from interference of other experiments. For the second case study, Emulab cannot support the wardriving experiment because its wireless devices are immovable. Though ORBIT can simulate the distance between wireless devices, an eavesdropper can scan wireless devices of other experiments due to the lack of isolation mechanism. MiNT and Mobile Emulab can support the wardriving experiment by deploying robots in an RF shielded room. However, the space for roaming is restricted. Only CMU testbed, ASSERT, and SWOONv2 can support device relocation and guarantee isolation for a wardriving experiment.

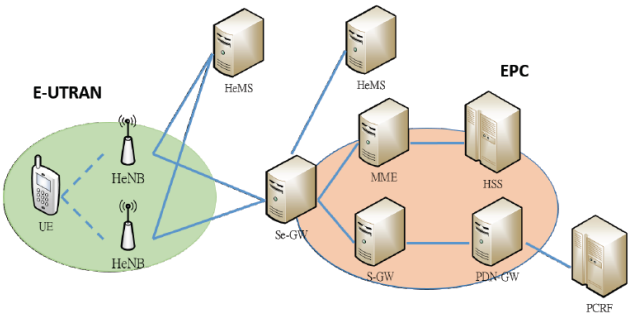


Fig. 3. LTE Architecture with eNB.

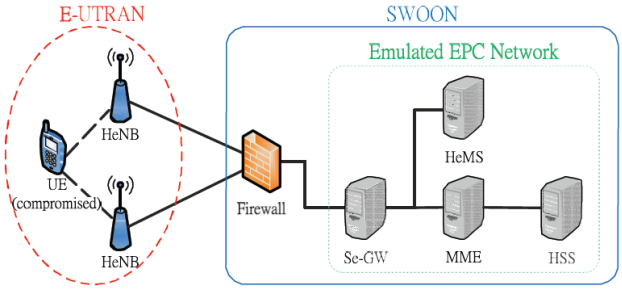


Fig. 4. Emulating LTE EPC network using SWOONv2.

CMU testbed, ASSERT, and SWOONv2 fulfill the isolation, fidelity and relocation requirements. But, considering of scalability, CMU testbed and ASSERT have a limitation on the scalability of an experiment, which depends on the number of attenuators installed on the FPGA board. Only SWOONv2 can support the creation of a larger experiment because the wireless transmission medium is emulated using VLAN technology, which can ideally support hundreds of devices and connections. This makes SWOONv2 a better choice for conducting wireless security analysis.

V. Discussion: Using SWOONv2 In Mobile Networks

LTE (Long Term Evolution) is a standard for high-speed data communication for mobile phones and devices. An LTE network can be divided to Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC) according to its transmission interfaces [15][16].

E-UTRAN, composed of several User Equipments (UEs) and Evolved Node Base stations (eNB), provides a radio link for a UE to connect to the EPC. A UE uses Uu interface (radio) to communicate with an eNB to attach to the LTE network. According to the radio coverage and transmission power, an eNB can be further classified into Macro, Micro, Pico, and Home eNB (HeNB) [17]. EPC is an all-IP network providing core services to LTE devices. UEs can obtain core service from EPC once they are authenticated via eNBs. If HeNBs are adopted in a LTE network, the HeNB management systems (HeMS) will also be included as part of the EPC network. Fig. 3 [18] shows the LTE network running with HeNBs.

In the specifications of LTE networking technology, many security mechanisms are designed for attack prevention [15][19][20][18]. To ensure that an LTE device can survive from common attacks, the device must be well tested before it is deployed to the real world. For this purpose, two kinds of testing instruments are offered in the existing market: simulation-based and emulation-based. For those simulation-based testing instruments (such as EAST simulators by EXFO [21], LTE NetEPC, LTE-HeNB-GW, LTE NetCell by Polaris Networks [22]), a tester can easily create test scenarios to evaluate the correctness of the protocols running on an LTE device. However, since no radio interface is provided, some physical attacks cannot be performed to evaluate the robustness of the device under test. For those emulation-based testing instruments (like ERCOM MobiPass [23], etc.), radio interfaces are provided and real life conditions can be reproduced and tested. A tester can build test scenarios only with the given functional modules. It is difficult for a tester to customize test scenarios (such as security tests). Since SWOONv2 can emulate a large network, SWOONv2 becomes a good candidate for testing LTE devices when running with the emulation-based LTE testing instruments. In the following paragraphs, we discuss how a testbed can be designed with SWOONv2 for LTE devices.

TABLE III
Comparison of SWOONv2 with other wireless testbeds

Testbed	Performance Measurement (Case I)	Wardriving Experiment (Case II)
EmuLab	Δ	X
ORBIT	Δ	X
MiNT-m	Δ	Δ
Mobile Emulab	Δ	Δ
CMU testbed	O	O
ASSERT	O	O
SWOONv2	O	O

O: Well supported. Δ: Partially supported. X: Not supported.

Similar to the way we used in Section IV, SWOONv2 can be used for testing LTE devices if an emulated radio interface is implemented in the WNIC of each mobile device. In case that no WNIC is implemented to emulate the Uu radio interface, a tester can also replace the WNIC with an emulation-based LTE testing instrument.

For example, to test the robustness of a LTE HeNB device against a DoS attack, a tester can emulate the EPC network using SWOONv2 and use an emulation-based testing instrument to emulate the compromised UEs, as illustrated in Fig. 4. In order to exhaust the network bandwidth for stopping normal UEs from obtaining core services, the emulation-based testing instrument (emulating the compromised UEs) can send numerous data via Uu interfaces to HeNBs and then to SWOONv2 (emulating EPC). With such a design, real life conditions can be reproduced and tested via the physical radio interfaces. Besides, by running a monitor in SWOONv2, the tester can mitigate the response time and understand how robust the HeNB device is when facing a DoS attack.

VI. Conclusion

SWOONv2 gives researchers a way to analyze the security of new wireless protocols or products without building a physical testing network by themselves. SWOONv2 adopts real machines to emulate network devices and allows a user to create arbitrary wireless network topologies without any hardware constraint. SWOONv2 provides strong isolation to shield test networks from interference and prevent test traffic from leaking to the public network, and such a characteristic allows a user to conduct high-risk wireless security experiments on SWOONv2. Although wireless packets are not transmitted through radio waves, SWOONv2 leverages the Ethernet broadcast protocol to broadcast wireless packets in an emulated network. SWOONv2 also simulates the distance between emulated devices and simulates signal

attenuation during transmission to support device relocation and experiment reproduction. Benefiting from the support of isolation, fidelity, relocation, repeatability and scalability, a user can validate his/her development against security requirements by conducting an experiment on SWOONv2. The high fidelity that SWOONv2 brings and the controllable testing environment supported by SWOONv2 can help reduce the complexity of test procedures and shorten the time to market for the new wireless protocols or products. SWOONv2 can also be applied to an LTE network for security tests. In the future, we plan to support more communication protocols for Virtual WNICs (such as RLC, PDCP, etc.) and adopt virtualization technology to improve the scalability of SWOONv2. Hopefully, SWOONv2 can become a new impetus for developing wireless technology.

Acknowledgements

This effort was partially supported by the Ministry of Education of Taiwan, the Taiwan Information Security Center (TWISC) Projects and Taiwan Ministry of Science Technology under Grants MOST 101-2219-E-009-016, MOST 101-2221- E-009-075 and MOST 101-2219-E-009-027.

References

[1] "The New Attack Vector: Applications." [Online]. Available:http://www8.hp.com/h20195/v2/GetPDF.aspx%2F4AA4-3092ENW.pdf

[2] S. Wahle et al., "Emerging Testing Trends and the Panlab Enabling Infrastructure," IEEE Communications Magazine, vol. 49, no. 3, pp. 167–175, 2011.

[3] G. Gibson et al., "PRObE: A Thousand-Node Experimental Cluster for Computer Systems Research," USENIX; login, vol. 38, pp. 37–39, 2013.

[4] M. Berman et al., "GENI: A Federated Testbed for Innovative Network Experiments," Computer Networks, vol. 61, pp. 5–23, 2014.

[5] T. Benzel et al., "Design of the Deter Security Testbed," USC Information Sciences Institute, University of California at Berkeley and McAfee Research, Tech. Rep., 2004.

[6] B. White et al., "An Integrated Experimental Environment for Distributed Systems and Networks," SIGOPS Oper. Syst. Rev., vol. 36, pp. 255–270, 2002.

[7] M. Ott et al., "ORBIT Testbed Software Architecture: Supporting Experiments as a Service," in Proceedings of Tridentcom'05, 2005, pp. 136–145.

[8] P. De et al., "MiNT-m: An Autonomous Mobile Wireless Experimentation Platform," in Proceedings of the MobiSys'06, 2006, pp. 124–137.

[9] D. Johnson et al., "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed," in Proceedings of INFOCOM'06, 2006, pp. 1–12.

[10] K. Borries et al., "Experience with a Wireless Network Testbed based on Signal Propagation Emulation," in Proceedings of EW'10, 2010, pp. 833–840.

[11] E. Nourbakhsh et al., "ASSERT: A Wireless Networking Testbed," in Proceedings of TridentCom'10, 2010, pp. 209–218.

[12] Y.-L. Huang et al., "SWOON: A Testbed for Secure Wireless Overlay Networks," in Proceedings of CSET '08, 2008, pp. 8:1–8:6.

[13] Y. Huang et al., "OSNP: Secure Wireless Authentication Protocol Using One-Time Key," Computers & Security, vol. 28, no. 8, pp. 803 – 815, 2009.

[14] "Linux WPA/WPA2/IEEE 802.1X Supplicant." [Online]. Available: http://hostap.epitest.fi/wpa supplicant/

[15] D. Forsberg, G. Horn, W.-D. Moeller, and V. Niemi, LTE security. John Wiley & Sons, 2012.

[16] M. Nohrborg, "LTE." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/98-lte

[17] J. Wannstrom, masterltefaster.com, and W. Keith Mallinson, "Heterogeneous Networks in LTE." [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/1576-hetnet

[18] 3GPP, "Security of Home Node B (HNB) / Home evolved Node B (HeNB)," TS 33.320.

[19] 3GPP, "3GPP System Architecture Evolution (SAE); Security architecture," TS 33.401.

[20] 3GPP, "3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses," TS 33.402.

[21] "EXFO Simulators-EAST." [Online]. Available: http://www. exfo.com/library/sales-marketing-resources/product-demos-interviews/bu1-exfo-nethawk-imulators-east-demo

[22] "Polaris Networks." [Online]. Available: http://www.polarisnetworks net/ henb-gw html

[23] "ERCOM MobiPass." [Online]. Available: http://www.ercom.com/fr/mobipass overview/4-26

Borting Chen received his B.S. and M.S. degrees in Electrical and Control Engineering from National Chiao Tung University, Taiwan in 2007 and 2009, respectively. He is now pursuing his Ph.D. degree in Institute of Electrical and Control Engineering of National Chiao Tung University since 2009. His research interests include network security, network testbed, cloud computing, embedded system and internet of things.

Yu-Lun Huang received the B.S., and Ph.D. degrees in Computer Science, and Information Engineering from the National Chiao Tung University, Taiwan in 1995, and 2001, respectively. She has been a member of Phi Tau Phi Society since 1995. She is now an associate professor in the Department of Electrical and Computer Engineering of National Chiao Tung University. She is now the Director of NCTU Center for Digital Content Production and Center for Continuing Education and Training. She has been serving the Secretary General of Taiwan Open Course Consortium since 2014. Her research interests include wireless security, virtualization security, embedded software, embedded operating systems, risk assessment, secure payment systems, VoIP, QoS and critical information infrastructure protection (CIIP), IoT Security, LTE Security, etc.

Tool, Technique, and Tao in Computer Security Education

Zhenkai Liang
National University of Singapore
liangzk@comp.nus.edu.sg

Jian Mao
Beihang University
maojian@buaa.edu.cn

Abstract - Computer security is a broad subject, covering topics in many subject areas of computer science. Moreover, this large body of knowledge is also actively changing with rapid advancement of computing technologies. In computer security education, students need the right methodology; otherwise they may feel at lost when facing the fast-changing security landscape. In this article, we discuss the approach and experience we had through several years of practice in computer security education. We advocate focusing on the fundamental concepts, to achieve deep understanding and flexible application of security knowledge through repeated reflection.

Keywords - Computer security education, Methodology

I. Educational Objectives

Educational objectives can be classified into several tiers [1]. Based on this taxonomy, Anderson et al. [2] revises the taxonomy of learning, teaching, and assessing as: *Remember*, *Understand*, *Apply*, *Analyze*, *Evaluate*, and *Create*. It is also a measure of how deep students understand knowledge: they may remember and understand *what* is the knowledge; they may learn *how* the knowledge can be applied to solve new problems; ultimately, they may know *why* the knowledge is created, and thus can analyze and evaluate existing knowledge and create new one. In the Chinese traditional philosophy, the levels of understanding (*what*, *how*, and *why*) are formulated as three corresponding levels of proficiency:

- **Tool (器):** Simple usage of basis of knowledge**.**
- **Technique (术):** Flexible application of knowledge.
- **Tao (道):** Fundamental understanding beyond the subject area.

Each individual has his/her unique view of the world and own way to learn new knowledge. Learning is a process to digest knowledge and transform it into an internal representation specific to each individual. Therefore, the only way to achieve understanding at the deepest level is through self-reflection, connecting and "compiling" knowledge into the inner representation of each individual.

II. Methodology for Learning in Computer Security

Education of computer security has been actively studied. Many efforts have focused on the curriculum and overall strategy in computer security education, such as the work by Bishop [3] and Moses-Petullo [4]. In addition to theory and abstract knowledge, hands-on practice is a critical aspect of computer security. To this end, Vigna [5] describes experiences in hands-on education of network security using live exercises on a testbed. Du et al. [6] developed a series of experiments for computer security education. They have demonstrated that such hands-on experiments play a key role in deepening students' understanding.

However, computer security is a fast changing field. Knowledge and tools are often outdated soon after, if not before, they are taught. It is difficult for curriculum and experiments to keep up with the rapid changes. To counter this challenge, we need to guide students to understand beyond the subject knowledge itself, to reach the reason why it is developed, and its connection with other knowledge.

Inspired by the principles of Chinese martial arts, whose ultimate goal is to build up the internal of a person's mind rather than to train the skills for physical movements, we believe the following principles from Chinese martial arts are inspiring to computer security education:

- **Countering changes with a constant principle (以不变应万变).**
- **Counter force with flexibility (以柔克刚).**

A constant principle leads to deep understanding of computer security knowledge, which will embed students with a set of methods to approach a large range of security problems. It also allows students to connect the knowledge well, so that they are flexible and quick in finding solutions.

What is the constant principle that enables flexible applications in computer security? We view computer security as *a new way of thinking*. For a new topic/area, the students should first focus on understanding how the system works. However, in addition to merely understanding it, they should think in a different angle as an attacker, to see how the system can be compromised, namely *break the system*. Next, it comes to understand the attack for how/why it works. The students can then think as a defender and see how to break the attacks to strengthen the system, namely *break the attack*. This will complete a cycle to bring the system with enhanced security,

which also starts off another cycle of "Understanding (System) - Breaking (System) - Understanding (Attack) - Breaking (Attack)." This learning cycle is summarized in Figure 1.

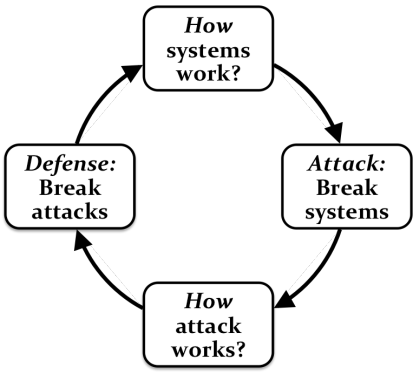


Figure 1. The learning cycle of computer security.

To equip students with this methodology, we need to guide them to practice; and more importantly, to reflect. In our teaching practice, we show how the cycle happens in many subjects. Taking the buffer overflow attack as an example, we start off by introducing the Intel architecture and illustrating how function call works. Then it is natural for students to "see" the buffer overflow attack by themselves. By analyzing the requirements of buffer overflow attacks, we show how defense solutions, such as stack protector and address space layout randomization (ASLR), break certain requirements of the attack. This process is illustrated in Figure 2.

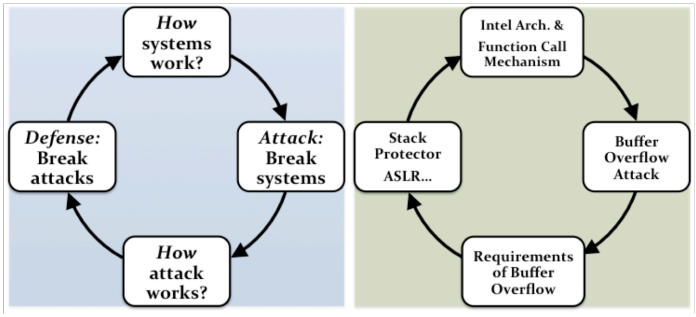


Figure 2. Mapping the learning cycle to buffer overflow attacks.

To help students reflect and grasp this methodology, we repeatedly show how to abstract this cycle from many topics in computer security, e.g., software security, web application security, and network security. In this process, they get the opportunity to strengthen their own understanding through the repeated exposures to this learning cycle in the context of many subjects.

III. Mission of University Education

In today's information age, many resources and services are readily available for training people, for example, the MOOC (Massively Open Online Courses). The strength of such resources and services is in *delivery* of knowledge, which is more about the tool and technique aspects of understanding. However, deep understanding at the Tao level needs individual reflection. So in our opinion, one of the unique strengths of university education is in individually guiding students to reflect, so that they can gain true understanding, enabling them to explore and innovate.

Concluding, we believed that one of the key missions of a quality university education is to engage students in actively thinking and reflecting. It can be summarized by the following Chinese proverb: 静思悟道 (Quietly thinking to reflect on Tao)

- **Quietly Thinking (静思):** The amount of information available to today's students is a blessing as well as a distraction. We need to teach students how to discard irrelevant or unimportant details so that one can focus on the principle.
- **Reflecting on Tao (悟道):** We should guide students on how to digest knowledge into their inner representation, achieving understanding at the deepest level.

Acknowledgments

We thank the editors for their comments to improve this paper. We thank Soo Yuen Jien, Ben Leong, Damith Rajapakse, and other colleagues for their inspiration and valuable comments.

References

[1] Benjamin S. Bloom. *Taxonomy of Educational Objectives: Handbook I: Cognitive Domain*. New York: David McKay, 1956.

[2] Lorin W. Anderson and David Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* New York: Addison Wesley Longman, 2001.

[3] Matt Bishop. *Teaching Computer Security.* In Proceedings of the Eighth International Conference on Information Security, 1993.

[4] Kyle V. Moses and W. Michael Petullo. *Teaching Computer Security.* In Proceedings of the ASEE Middle Atlantic Section Meeting, 2014.

[5] Giovanni Vigna. *Teaching Hands-on Network Security: Testbeds and Live Exercises.* In Journal of Information Warfare, vol. 2, issue 3, pp 8-24, 2003.

[6] Wenliang Du, Karthick Jayaraman, and Noreen B. Gaubatz. *Enhancing Security Education with Hands-on Laboratory Exercises.* In Proceedings of the 5th Annual Symposium on Information Assurance (ASIA '10), 2010.

Zhenkai Liang is an associate professor at School of Computing of National University of Singapore. His main research interests are in system and software security, web security, mobile security, and program analysis. As a co-author, he won six best paper awards. He also won the Annual Teaching Excellence Award of NUS in 2014 and 2015. He received his Ph.D. degree from Stony Brook University, and B.S. degree from Peking University.

Jian Mao is an assistant professor at the School of Electronic and Information Engineering of Beihang University. Her main research interests include cloud security, web security, and mobile security. She won the First-class Teaching Award from Beihang University in 2012 and 2014. She received her Ph.D. degree and B.S. degree from Xidian University, China.

CTF: Alternative Training for Offensive Security

Chung-Kuan Chen
National Chiao Tung University
ckchen@cs.nctu.edu.tw

Shiuhpyng Shieh, Fellow of the IEEE
National Chiao Tung University
ssp@cs.nctu.edu.tw

***Abstract*—Offensive security is a concept of defense based on adversary's mindset. Traditional security education is insufficient for offensive security. Most security courses are designed from the defender's perspective, and have a big gap from the real world problems in practice. The emerging contest Catch-The Flag (CTF) can complement the traditional education for security training. In a CTF contest, competitors will try to solve security problems that are verisimilar to real world problems. The CTF infrastructure also provides the platform to sharpen competitors' security skills. There are three types of CTF competition: Jeopardy, Attack and Defense, and King of the Hill. While Jeopardy can cover a wide range of knowledge of skills, such as pwn, reverse engineering, web, forensics, and cryptography. Attack and Defense competition focuses on the entire vulnerability life cycle. In this type of competition, competitors are given green light to attack each other. As a typical example, the target program is first analyzed. If vulnerabilities are discovered, an attacker can develop exploits to compromise other teams' machines. Meanwhile, to counter the same attack, the vulnerabilities of their own system should be patched. King of the Hill is the variant of Attack and Defense. The longer the competitors can take over the system, the more score they can get. With these different types of CTF competition, the offensive security skills can thus be polished. To encourage automating generation of the defensive system, the DARPA recently launched the Cyber Grand Challenge aimed for the first CTF competition played solely by machines.**

***Keywords*—Security training, Jeopardy, Attack and Defense, King of The Hill, Catch The Flag**

I. Offensive Security Training

Offensive security, complementary to the traditional defensive security, is the concept of defense based on the comprehension of adversary's mindset. Consequently, security experts can identify the weaknesses for potential compromise in the whole system, and then accordingly proper defense mechanisms can be chosen and deployed. Offensive security awareness is important even for system and network administrator. Every administrator should learn the concept of offensive security to certain degree. In the software development cycle, programmers should understand common weaknesses that are easy to attack by an adversary. Software programmers should avoid writing programs with weaknesses, such as, buffer overflow, and SQL injection. Similarly, these weaknesses can also be avoided in the code review step if the reviewer understands the intrusive approach taken by the adversary.

For security experts, the importance of offensive security drastically increases in recent years. For penetration testing of a software program, there may exist hundreds of entries potentially vulnerable to attacks. Testing these entries individually is very time consuming, and without the knowledge of adversary's method the test itself may fail. To effectively discover system vulnerabilities, the penetration testers should be familiar with adversary's penetration methods.

To achieve offensive security, education for offensive thinking is the first step. However, adversary's thinking is difficult to learn through traditional courses and education. The thinking of an attacker is opposite to that of a traditional software developer. For the developer, the software construct process starts from software design, followed by program development, and then software testing. In contrast, the adversary hacking starts from software testing to discover vulnerabilities, followed by reverse engineering to understand the software, and finally development of exploits. As a result,

traditional education in compliance with the software construct process fails to cover offensive security. Furthermore, security-related courses only focus on defense rather than offensive security.

The other issue is practice. Similar to traditional software development, practice makes perfect. Adversary skills also require significant effort in practicing. Different skills may be adopted in different situations. The traditional security education often focuses on basic concepts and theory. Little practice is involved in the course. Therefore, the gap between the course and real world problems exists. In addition, the lack of practice is also a problem for offensive security education. Due to the intrusive behavior of hacking skills, it is not a good idea to conduct exercises online in a public network where real systems may be corrupted and malicious behavior is forbidden.

II. CTF: THE WORLD WIDE GAME FOR HACKERS

Capture the Flag (CTF) is a promising solution to offensive security education and talent discovery. In a CTF contest, competitors should think as a hacker and break security problems. Security-related problems are designed and announced to competitors by CTF organizers. The competitors' goal is to find the flag, a string crafted in a specific format, hidden in the problems via some security exercise.

The first CTF contest, Defcon CTF, started in 1996. Until now, Defcon CTF is still the most important contest in the world. Every year, hundreds of teams participate in the Defcon qualification for the chance to be part of the final contest in Las Vegas. In the Defcon final, a few qualified elite teams compete in the live, face-to-face environment to pursue the championship.

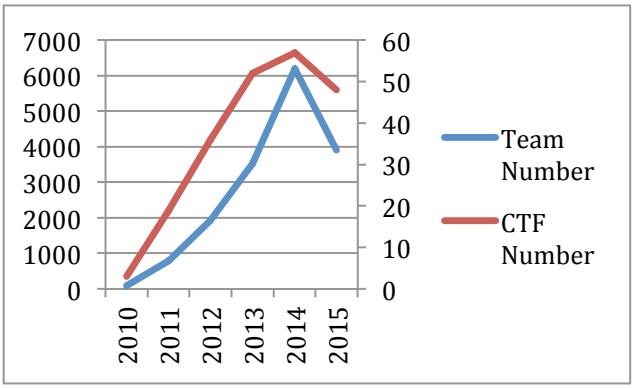


Figure 1 *Trend of CTF and Teams*

Besides Defcon CTF, there are many other CTFs [1-20] held around the world, such as SECCOM held in Japan, CodeGate in Korea, RuCTF in Russia, HITCON in Taiwan, and 0CTF in China.. This indicates that CTF gradually becomes a critical platform for offensive security training. All of these CTFs attract hundreds of teams to participate.

XCTF is the CTF tournament organized in China. In the tournament, a number of CTFs are held at different universities. The front-runners with higher overall ranking are qualified to compete in the XCTF final contest. Among all the CTFs in China, 0CTF and BCTF are the world-class competitions. Other CTFs in XCTF are mainly for students in China. CTF teams around the world can still participate in these CTFs. With this high frequency of CTF contests, both experts and students have remarkable opportunities to learn and polish their security skills.

CTF contests are not solely designed for security experts. Different CTFs may target at competitors at different levels. As a beginner of CTF, Backdoor CTF, ASIS CTF, Hack.lu CTF are good choices to enter the world of hacking. On the other hand, DEFCON, PlaidCTF, CodeGate and Ghost in the Shellcode CTF give more advanced and challenging problems to the competitors. Due to the wide range of difficulty, one can find the CTFs suitable for them to learn security skills.

To attract more students to learn security, some CTFs are dedicated to students. Among them, CSAW CTF is designed for undergraduate students to learn skills during the contest. It was the biggest contest in 2014, and more than 18,000 competitors from 75 countries around the world attended. In addition, picoCTF and HSCTF further reach out to high school students. These CTFs help high school students get exposed in the exciting new area.

Even though all the CTF contests aim for security education, each CTF may have its individual goal. For example, PHDays (Positive Hack Days) CTF is designed to mimic the real-life conditions where the contemporary vulnerabilities of information system are adopted as the contest problems. Moreover, depending on the contest scenario, the underlying infrastructure may change over time. In this way, competitors can practice to solve real-world problems for a variety of infrastructures. iCTF, the UCSB International Capture The Flag, targets on building a distributed, world-wide security contest for more students to join and learn attack and defense skills. The iCTF framework is also published for others to establish their own CTF contest. Thus, it may ease the difficulty to hold a CTF contest.

One important feature of CTF is gamification which makes the contests more interesting and attractive to students. Some CTFs try to make their contests more fun. PlaidCTF 2012 constructs the contest as the RPG (Role-Playing Game) game, shown in Figure 2, where competitors play the role as an adventurer to execute a mission that is indeed a security problem. Ghost in the shellcode CTF includes problems hosted on Pwnie Island since 2014. Pwnie Island is the first personal, open-world MMORPG (Massively Multiplayer Online Role-Playing Game). In the game, a competitor should complete the missions impossible unless game hacking techniques are used.

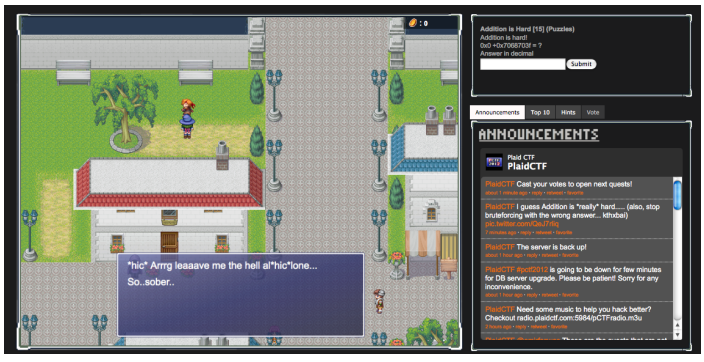


Figure 2 *RPG Style CTF in PlaidCTF 2012*

According to the authoritative CTF ranking site CTFtime, about 100 CTF contests are recorded and 16335 teams has registered till 2015. These CTF teams come from 64 different countries around the world. The trend of growing number of teams and CTFs is demonstrated in Figure 1. The blue line in the figure denotes the total number of teams registered to

CTFTime over the years. And the red line indicates the total number of CTFs held each year. As the figure demonstrates, the total number of teams participated in CTF grows from 90 in 2010 to 6197 in 2014, while the total number of CTFs also increases from 3 in 2010 to 57 in 2014. Note that, the data collected for 2015 is only up to August, and therefore the decrease in 2015 does not indicate the actual number.

III. Different Types of CTF

There are three types of CTF contests, Jeopardy, Attack and Defense, and King of the Hill. Consider the last two CTF types with the requirement of a stable network. They are mostly held in a closed environment and only a limited number of teams can join. Therefore, many CTFs hold Jeopardy-style CTF in the qualification round. Only the best teams can participate in the finial round of CTF that is either in Attack and Defense or in King of the Hill style.

A. Jeopardy

Jeopardy shown in Figure 3 is the most common CTF type. Just like the TV shows with the same name, competitors will encounter several problems in different disciplines. In general, Jeopardy CTF includes the problems on pwn, reverse engineering, web security, forensics, and cryptography. Each problem has a score that indicates its difficulty. Once a competitor solves the problem, the associated score is earned. As a proof of problem solving, the flag hidden in the problem is discovered and submitted to the organizer as evidence.

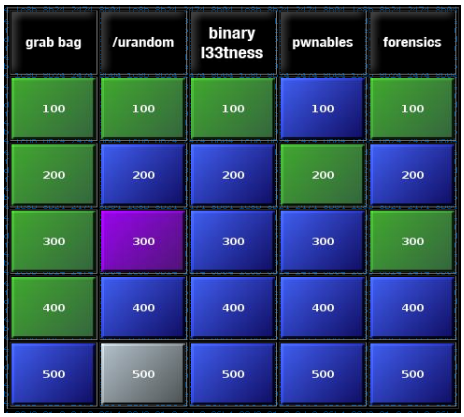


Figure 3 *Scoreboard of Jeopardy CTF, Defcon CTF*

1) Pwn

In the pwn discipline, a vulnerable executable and the information of a remote server are given. With the executable, a competitor can analyze the executable offline and discover vulnerabilities such as buffer overflow, use-after-free, and format string. The exploit—an attack program based on the vulnerability—can be developed to steal the flag from the remote server.

2) Reverse

The reverse engineering problem often provides an executable with the flag hidden by the designed algorithm. With disassembly and decompiler tools, competitors need to understand the algorithm's logic and unveil the key. For example, given the user validation program with unknown registration code, to generate registration code, the final flag can be unveiled after understanding the algorithm.

3) Web

Similar to the pwn problem, a remote web server with vulnerabilities is given. The difference is that the program is not given in most cases. Therefore, the black box analysis is needed to discover web vulnerabilities such as, SQL injection, cross-site scripting (XSS), and command injection.

4) Forensics

In forensic problems, the flag is hidden in the object given. The target objects vary from network packets, document files to disk images. Realizing the principle of system can help solve the problems. For example, a disk image is given, but the flag is inside a deleted file. Therefore, knowledge of the file system should be understood, and some forensic tools can be used to recover the deleted file.

5) Cryptography

Cryptographic problems usually contain the encrypted flag. While encryption, some weaknesses are involved in the cryptographic system. Competitors can therefore break the encryption algorithm and find the flag. Weak cryptographic algorithms, such as Vigenere cipher, can be broken by frequency analysis and brute force. Short key encryption and insecurely chosen small prime numbers in RSA are some examples of the cryptographic problems.

B. Attack and Defense

Attack and Defense CTF is a verisimilar contest in contrast to Jeopardy. The competitors are put into a closed environment, trying to attack each other. The server, a k.a. gamebox, is given to each team. Just like the real environment, several vulnerable services are deployed in the gamebox. The service can be any kind of network services, such as vulnerable website, and socket program. Therefore, the basic requirement for a competitor is to maintain service availability and at the same time to compromise other team's service.

To exercise the attack and defense skills, the whole life cycle of vulnerability, from vulnerability discovery to exploit development and service patch, is involved in the contest. Due to the identical environment given to each team, competitors need to analyze their service to discover the vulnerabilities. Then, the exploit code can be developed to steal the flag associated with each service as the evidence of successful compromise. At the same time, a competitor should protect his service from being compromised. Hence, patching the vulnerability is also an urgent task.

Attack and Defense CTF is a zero-sum game. If a team gains some score, the other team will lose the same score. Each service has three statuses: service alive, service down and service compromised. The minus points will be given for service down or service compromised. In the situation of service compromised, the losing score will be evenly given to the teams that successfully compromise this service. If a service is down, the score will be also evenly distributed to other teams whose service are still alive. Hence, the overall score remains the same during the contest.

As the real world, time is an important factor for the Attack and Defense CTF. The duration of competition is divided into several rounds, e.g. 5 minutes a round. In each round, the flag of each service is updated. Only the flag of current round is worth the score. Therefore, the earlier an exploit is developed, the more flags and higher score will be received. On the other

hand, patching the service earlier can mitigate the score loss in the early phase.

Not only the security skills can be trained in the Attack and Defense CTF, the system administration skill also plays an important role. Permission setting is the basic skill needed. With misconfiguration, the adversary can compromise the service. Monitoring system status to discover hidden backdoors and cleanup backdoor process is another important administrative work. Network packet analysis is not only used for defense but also used for attack. After understanding the attack method via network packet analysis, replay attacks may be possible by constructing other teams' attack exploits.

In contrast to the Jeopardy CTF held online for competitors around the world to participate, the Attack and Defense CTF is often held in a designed and controllable environment due to its complexity. Therefore, only a limited number of teams can participate in the contest.

C. *King of the Hill*

King of the Hill is the CTF type derived from the Attack and Defense CTF. King of The Hill is also held in the closed environment. Each team also needs to attack other teams and patch vulnerable services. The key concept of King of the Hill is keeping the control over the system as long as possible. The limited number of services will be provided to competitors. All the competitors can access each service and discover its vulnerabilities. After competitors get control of the service, they can try to patch the service and prevent other teams to take it over. The longer a competitor controls the service, the higher score will he receive.

King of the Hill CTF is verisimilar to the real world situation where only one team can take control of a service. Just like website defacement, each team should put their identity in the service to prove the successful compromise. Different from the Attack and Defense CTF where every team successfully compromising the service can get score, only the team remains in control of the service can gain the score. Other team does not get any score even if they compromised the service.

To reflect the real situation, the infrastructure in the King of the Hill consists of services widely used in the real world, such as LAMP server, AD server, database server and web server. Therefore, an attacker should polish his attack method for each service and at the same time gets familiar with operations to patch the service.

IV. CTF as Education for Practical Offensive Security Skill

To complement traditional security courses, CTF can educate students with verisimilar practices and real world problems. For the Jeopardy style CTF, competitors can learn a wide range of security problems, from web security to reverse engineering and software exploitation. In contrast, the Attack and Defense CTF focuses on the life cycle of vulnerability, thereby learning skills from vulnerability discovery to exploit development and software patching.

It takes time and effort to deploy new techniques, such as SDN and IoT, in the real world. However, CTF is based on the emulated environment, hence new techniques and platforms can be introduced more easily. For example, DEFCON final already used ARM architecture instead of x86 as the attack and defense architecture for several years. In Boston Key Party CTF, problems about SDN are given. Hence, not only practical but also advanced techniques can be exercised in the contests.

CTF can also be the playground for new attack methods. While solving the problems, competitor can adopt advanced attack to bypass some defense mechanism. Several advanced attack methods already appeared in CTF contests. For example, SROP (SigReturn Oriented Programming) is the attack method proposed in S&P 2014. And several problems related to research topics, such as symbolic execution and sandbox breaking, are designed as problems in CTFs.

Moreover, experience and skills learned from CTF can help competitors realize and deepen their research. DARPA recently launched a project called CGC (Cyber Grand Challenge). CGC is aimed to be the first CTF played solely by machines. In the contest, competitors have to automate the process of vulnerability discovery and software patching. This is just one of the examples that shows the impact of CTF experience to the future research.

In summary, CTF can complement traditional security education to enhance offensive security training. Putting the competitors into the CTF and facing real problems can bridge the theory and the practice. While solving the problems, both offense and defense skills can be practiced. The infrastructure used in the CTF contest can be the best playground for competitors to exercise their security skills.

Acknowledgements

This work is supported in part by Ministry of Science and Technology, Ministry of Education, TWISC, ITRI, III, iCAST, TTC, HTC, D-Link, Trend Micro Inc., Promise Inc., Chungshan Institute of Science and Technology, Bureau of Investigation, and Chunghwa Telecomm.

References

[1] DECON CTF, https://www.defcon.org/html/links/dc-ctf html

[2] SECCON CTF, http://ctf.seccon.jp/

[3] CodeGate CTF, http://ctf.codegate.org/

[4] RuCTF, http://www ructf.org/

[5] 0CTF, https://ctf.0ops.sjtu.cn/

[6] HITCON CTF, http://hitcon.org/2015/

[7] XCTF, https://time.xctf.org.cn/

[8] BCTF, https://bctf.cn/

[9] Backdoor CTF https://backdoor.sdslabs.co/

[10] ASIS CTF http://asis-ctf.ir/home/

[11] Hack.lu CTF http://2015.hack.lu/

[12] PlaidCTF http://play.plaidctf.com/

[13] Ghost in the Shellcode http://ghostintheshellcode.com/

[14] CSAW CTF https://ctf.isis.poly.edu/

[15] picoCTF https://picoctf.com/

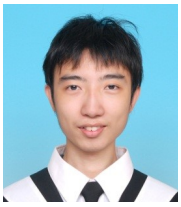
[16] HSCTF http://hsctf.com/

[17] PHDays CTF http://www.phdays.com/program/contests/

[18] iCTF http://ictf.cs.ucsb.edu/

[19] Pwnie Island http://pwnadventure.com/

[20] CTFTime https://ctftime.org/



Chung-Kuan Chen is a PhD Candidate in the Department of Computer Science, National Chiao Tung University (NCTU), Hsinchu, Taiwan. Contact him at ckchen@cs nctu.edu.tw.

Shiuhpyng Winston Shieh is a distinguished professor and past chair of the Department of Computer Science, NCTU; and the Director the Taiwan Information Security Center at NCTU. He is an IEEE fellow and ACM Distinguished Scientist. Contact him at ssp@cs.nctu.edu.tw.

Submission Instructions

Authors should use the designated IEEE Reliability Manuscript Central Website to summit their papers. Please refer to the following steps to submit your papers:

1. Login to IEEE Reliability Manuscript Central. If you have no account, sign up for one.
2. Click "Authors: Submit an article or manage submissions".
3. Please click "CLICK HERE" at the bottom of this page, and you will be brought to the five-step submission process.
4. You need to 1) choose the section that you are going to submit your paper to; 2) complete the submission checklist; 3) enter the comments for the editor, which is optional; 4) save and continue.
5. If you have any supplementary files, please upload them in step 4.

Manuscript Types

Manuscripts for regular issues fit within the scope of the magazine, but are not intended for a special issue. Special issue manuscripts cover a specific topic scheduled on our editorial calendar. Please select the appropriate issue (manuscript type) when uploading your manuscript. For more information and to see upcoming special issue topics, see our Editorial Calendar at http://rs.ieee.org/reliability-digest/author-guidelines.html.

Typing Specifications

The manuscript should be written in Times New Roman in a double-column format. The typical length of the submitted manuscript is 4 single-spaced pages. The text portion of the manuscript should be in 10-point font and the title should be in 24-point font, bold.

Manuscript Length

The typical length of the submitted paper is 4 pages, including text, bibliography, and author biographies. Please note that proper citations are required.

Illustrations

The illustrations in the articles must be cited in the text and numbered sequentially. Captions that identify and briefly describe the subject are needed as well. In order to avoid dense and hard-to-read illustrations, graphs should show only the coordinate axes, or at most the major grid lines. Line drawings should be clear. To prevent potential layout problems from happening, related figures described within the same section of text should be grouped together as parts (a), (b), and so on.

References

All manuscript pages, footnotes, equations, and references should be labeled in consecutive numerical order they are mentioned in the text. Figures and tables should be cited in text in numerical order.

Biographical Sketch

A brief biographical sketch should contain the full title of the paper and complete names, affiliations, addresses, and electronic mail addresses of all authors. The corresponding author should be indicated.

Please provide a short biography and a picture for each author of your paper at the end of your paper. The short biography should contain no more than 150 words

Copyright

IEEE Reliability Society owns the copyrights of the articles published in Reliability. If you wish to reproduce the copyrighted materials, please contact us and seek the permissions. The contents on this website can be referenced with proper citation.

Special Issue Proposal Submissions

For a special issue in Reliability, experts are welcome to serve as our guest editors. To know more information, please contact Editor-in-Chief on Reliability, Shiuhpyng Shieh: ssp@cs nctu.edu.tw.