AUTOMOTIVE ETHERNET SECURITY TESTING

ixia
A Keysight Business

TECHNICAL GUIDE

Highlights

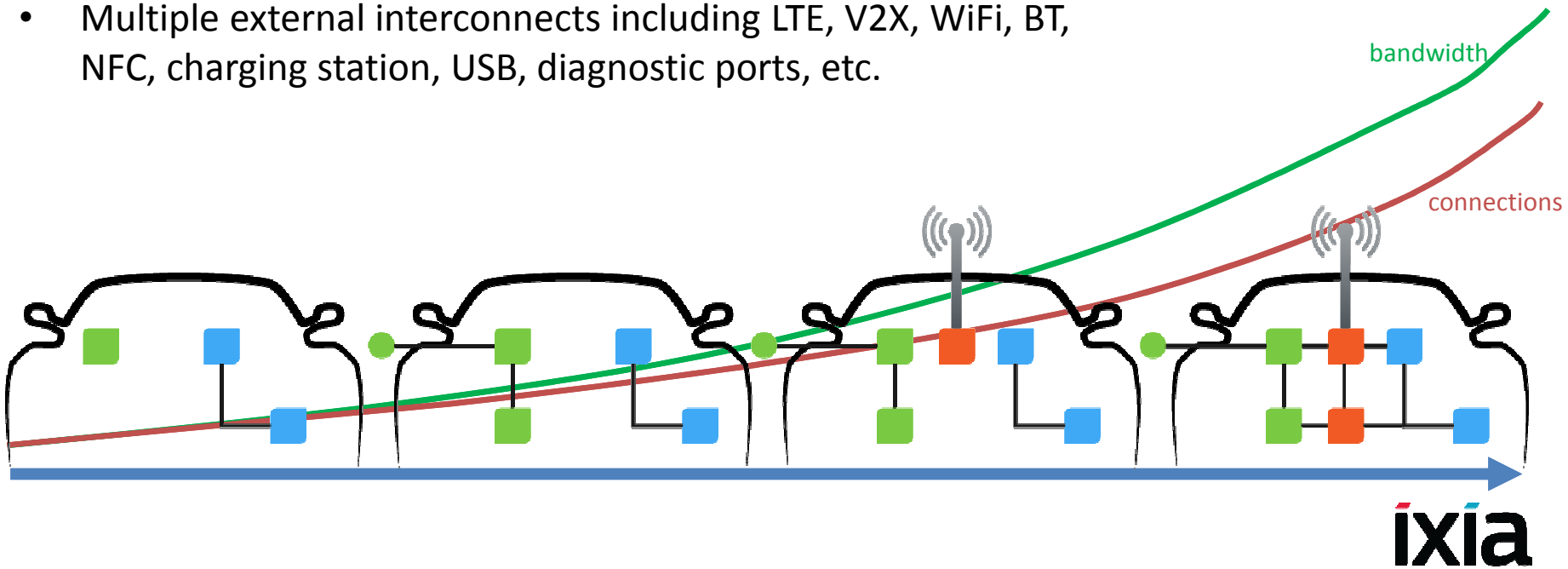
- Cars are evolving
- Types of attacks
- Car hacking is a real threat
- Why is security important in automotive?
- What is a vulnerability?
- Layered approach to security
- Conformance testing
- Testing for known vulnerabilities
- How fuzzing test cases are written
- Running tests in the automotive Ethernet environment
- Fuzzing: Testing for unknown vulnerabilities
- Testing for stability and resiliency

Learn more at: www.ixiacom.com

For more information on Ixia products, applications, or services, please contact your local Ixia or Keysight Technologies office. The complete list is available at: www.ixiacom.com/contact/info

Cars are evolving

- Number of ECUs, sensors, and interconnects is growing
- Moving to Ethernet networks utilizing TCP/IP and other well-known protocols
- ECUs in different car domains will be interconnected with Ethernet backbones
- Multiple external interconnects including LTE, V2X, WiFi, BT, NFC, charging station, USB, diagnostic ports, etc.

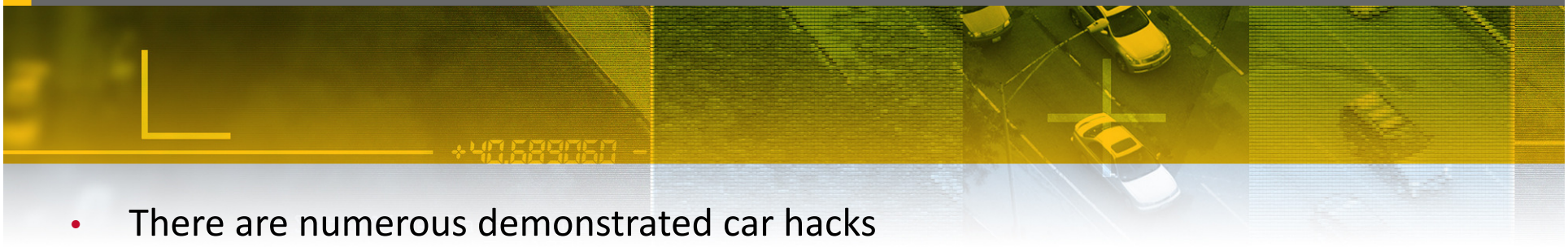


ixia

Types of Attacks

ixia

Car Hacking is a real threat



- There are numerous demonstrated car hacks
  - Most documented attacks are physical attacks via CAN
  - New attacks are possible via Automotive Ethernet networks
- There are hacking manuals / books for many cars
- Fewer cost barriers
  - Automotive Ethernet connectivity devices are relatively cheap
  - Free protocol analyzers available
- Fewer technology barriers
  - Use of TCP/IP and Ethernet entices hackers from the IT domain to try to hack cars
- Automotive firmware is not updated quickly/frequently
  - Once a weakness is known, it is hard to prevent attacks

ixia

Why is security important in automotive

- Potential results of car hacking
  - Theft
  - Loss of privacy
  - Recalls or upgrades for insecure components
  - Damage to the vehicle
  - Bodily injury
  - Loss of trust by the consumer
  - Loss of revenue to the OEM

ixia

What is a vulnerability

- A vulnerability is a weakness which makes the system susceptible to unauthorized access or malicious behavior.

- Well-known (published) vulnerabilities are the #1 way that hackers gain access to a computer

- Common Vulnerabilities and Exposures (CVE) database
  - Database used by all security companies
  - Lists publicly known vulnerabilities
  - Some are specific a platform, but many are generic
  - Currently over 50,000 vulnerabilities identified
  - Use of CVE is standardized by the ITU, NIST, etc.

ixia

Example of a Vulnerability

How does a stack based buffer overflow work?

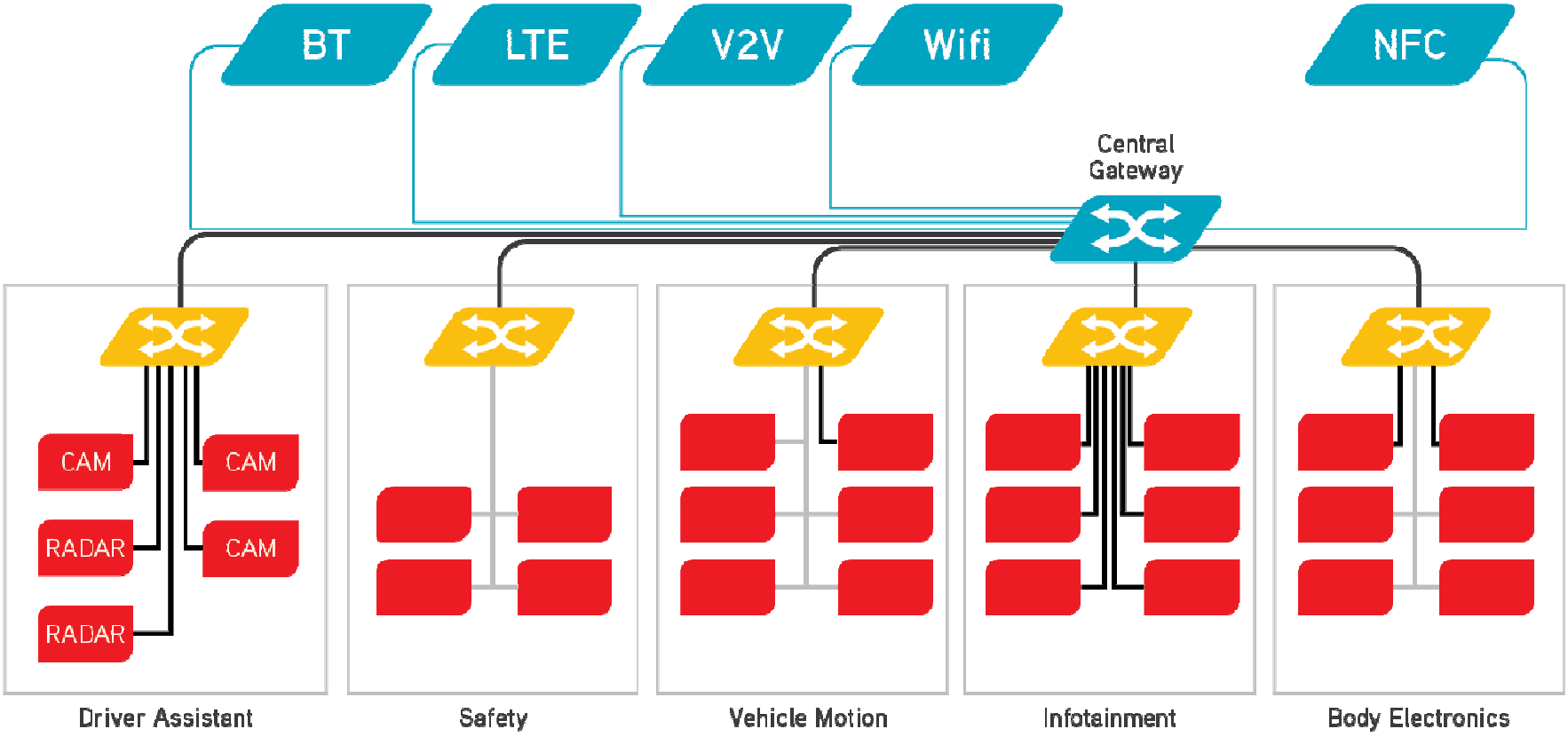
Give this program 100 A's…

```
void foo(char *user_str)
{
char local_str[64];
strcpy(local_str, user_str);
}
int main(int argc, char *argv[])
{
 if(argc!=2)
       { printf("usage: %s <in_string>\n", argv[0]); return 1; }
foo(argv[1]);
return 0;
}
```

```
Registers (FPU)
EAX 00000000
ECX 00320FB4
EDX 00414141
EBX 7FFDD000
ESP 0012FEEC ASCII "AAAAAAAAAAAAAAAAAAAA
EBP 41414141
ESI 00000A28
EDI 00000000

EIP 41414141

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BDEC 01050104 002E0067
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P U O Z D
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1
```

ixia

Automotive Network Diagram



ixia

Vulnerabilities, Exploit, Malware, and Viruses

- Definitions
  - Security vulnerability: a weakness which makes the system susceptible to an attack
  - Exploit: an attack that uses the vulnerability
  - Malware: any software that runs on a CPU and performs unwanted tasks
  - Virus: a type of Malware that is designed to spread to other CPUs (typically using a known vulnerability)
- Anti-Virus software detects instances of viruses and malware, but does not detect (or fix) the underlying vulnerability
- Finding and fixing vulnerabilities in every layer is critical to preventing attacks

Layered approach to security

- Layered security is “the practice of combining multiple mitigating controls to protect resources and data” *–Wikipedia*
  - In order to protect against a broad range of attacks, using multiple strategies is more effective
  - If one layer is bypassed, other layers may offer protection
- When breach occurs, other subsystems should remain resilient to the attack
  - For example, a Denial of Service attack should not affect the braking function of the car
- Each layer needs to be tested.

ixia

Conformance Testing

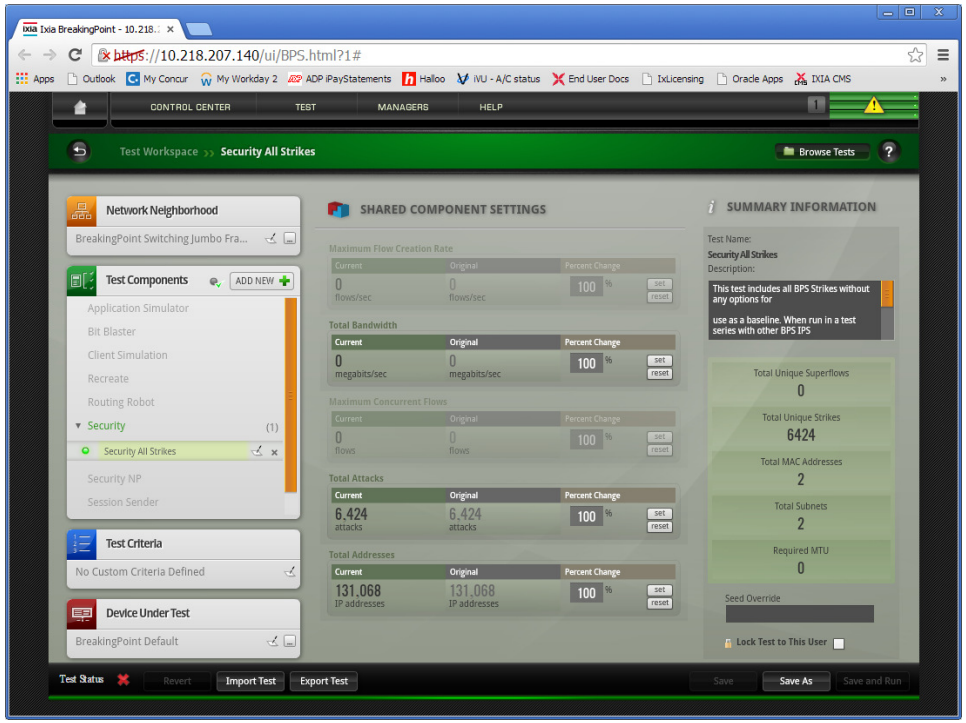
- Proper architecture, design, and implementation is the first line of defense
- Conformance testing is the second line of defense

- Conformance testing validates that the implementation conforms to the design specifications and standards
    - For every requirement in each applicable specification, there needs to be at least one conformance test
    - The robustness & security requirements must be considered when defining test cases
    - Both positive & negative tests are needed to ensure security robustness

ixia

Testing for known vulnerabilities

- There are multiple tools to test for known vulnerabilities
  - The tools vary in their approaches and coverage
  - No single test tool covers all vulnerabilities
- The tools run tests that exploit the vulnerability in a similar way to a hacker



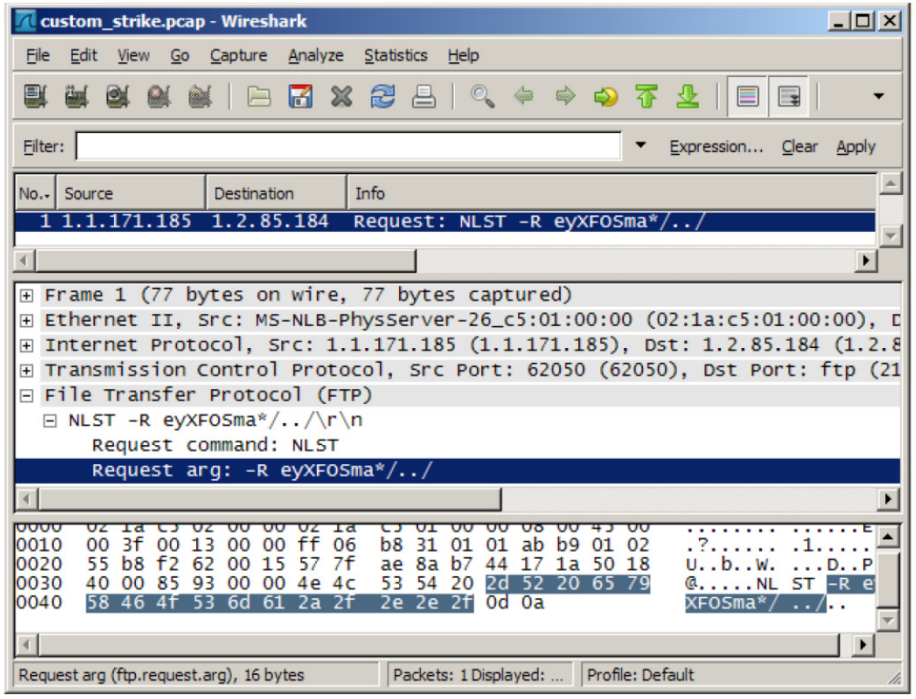
Screen from a tool running a test using 6000+ attacks

ixia

How Fuzzing Test Cases are Written

- Test cases are typically provided by the test tool vendor, but custom tests can be created

- Exploit is developed based on vulnerability description
- Test run and Ethernet traffic is captured
- Based on the capture, test is defined in the test tool
- Test is verified against a vulnerable system



Sample capture of an exploit

ixia

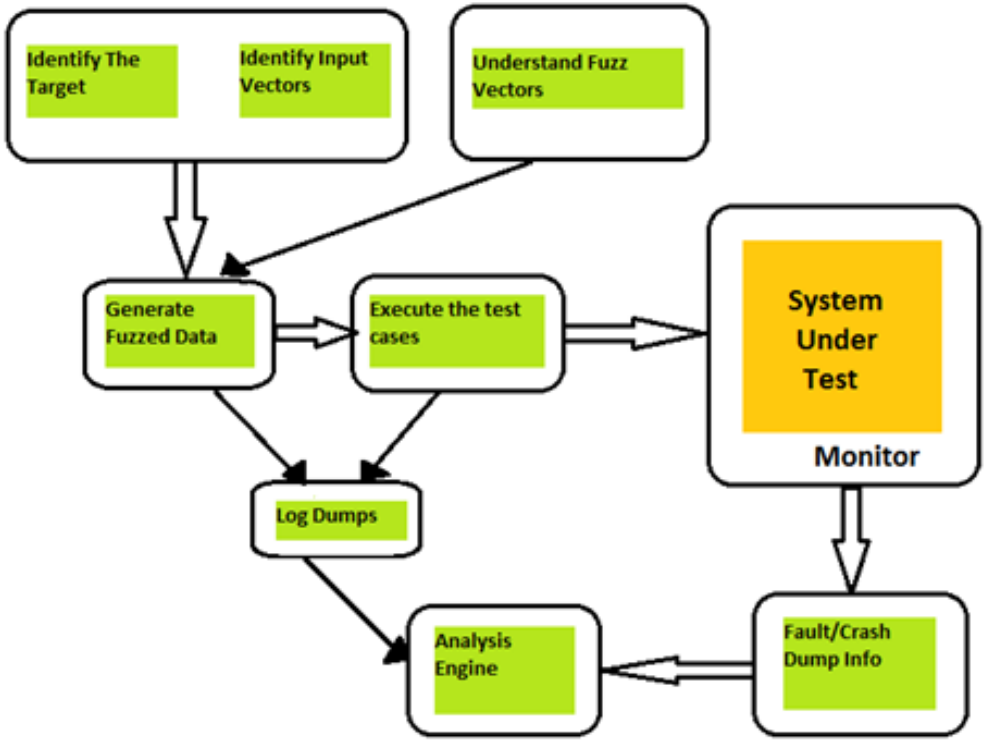
Running in the Automotive Ethernet Environment

- Configure the test based on the Automotive Network
  - Test needs to be configured for the addresses & protocols used in the devices under test
- Some test use features not used in the automotive domain
  - ECUs only implement protocols & features required for operation
  - Tests need to be modified based on the ECU configuration
- Some tests need modifications for automotive-specific protocols
  - Examples include SOME/IP and diagnostics over IP
- Every system component (ECU, network infrastructure device, gateway, operating system, etc.) needs to be individually tested.

ixia

Fuzzing: Testing for Unknown Vulnerabilities

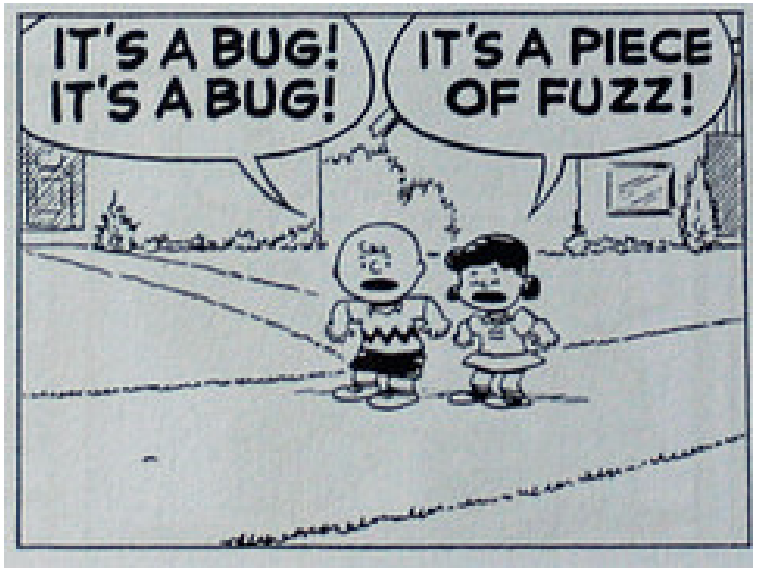
- Fuzzing is a technique where invalid, unexpected, or random data to the inputs of a computer program
  - Best technique to find unknown vulnerabilities such as buffer overflows or error handling
  - For Ethernet testing, specific fields of a protocol are “fuzzed”
  - In parallel, the system is monitored for faults (to identify vulnerabilities)



ixia

Fuzzing: Testing for Unknown Vulnerabilities

- Why fuzzing
  - Good architecture, design and review is a good first defense against unknown vulnerabilities
  - Even the best review will not find all vulnerabilities.
  - It is estimated that for every 1000 lines of well-written code, there is approximately 1 vulnerability
  - By generating values that software does not expect, fuzzing finds mistakes that are typically not found by conformance & performance tests (which focus on functional use cases)



ixia

Using Fuzzers

- For best results, use a protocol-aware fuzzer and configure the fuzzing based on an understanding of the protocol and implementation
- Start with white box testing at the function or component level
  - Fuzzing can be focused based on knowledge of the code
  - Easier to detect malfunctions with white-box testing
  - Use Code coverage analysis: condition coverage reveals what cases have been exercised by the fuzzing tool
- Work up to testing at the system level using black-box methodology
  - Repeat and expand on the tests run at the lower levels
- Run fuzzers for a long time
  - Due to the random nature of fuzzing, finding vulnerabilities may take a long time
- To increase coverage, use different fuzz algorithms and fuzz different fields

ixia

Testing for Stability & Resiliency

- When an attack is attempted, it can have an effect on other components (even if the attack is unsuccessful)
  - the amount of network traffic or function calls may increase
  - response times from some components may increase
  - ability to communicate with external devices may deteriorate
  - memory consumption may go up
- It is important to evaluate all layers and components in the system to assess their stability and resiliency under such conditions.

ixia

Testing for Stability & Resiliency

- In order to test for Stability and Resiliency, several methodologies are used
    1. Firewall testing is used to validate a gateway or firewall configuration and evaluate performance
    2. DDoS mitigation testing evaluates a gateway or firewall operation under a DDoS attack and validates that malicious traffic is blocked
    3. Stress testing is used to drive components beyond normal operational capacity to observe how the system functions
    4. Resiliency testing is used to validate operation under degraded or failure conditions (i.e. a sensor failure)
    5. Impairment testing is used to validate performance when communication is impaired (typically testing with delayed, dropped, or erroneous packets)
    6. Functional and performance tests need to be run on the security components under attack conditions (as attack conditions should be part of the “normal” testing for security components)

ixia