

## Re: Pentesting tool – Commercial

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/securityfocus/pen-test/2008-03/msg00022.html>

---

- *From:* Trygve Aasheim <[trygve@xxxxxxxxxxxxxx](mailto:trygve@xxxxxxxxxxxxxx)>
  - *Date:* Tue, 04 Mar 2008 20:54:45 +0100
- 

This might be a bit hard for you to understand, I see that, but just trust me – ok?

There is a world outside of web servers and web applications. There are tests that needs to be done outside the scope of owasp, and there are companies with more complex systems than those of auction sites.

Parts of this world contains servers that performs different tasks like backup, store databases, process data, pass mail etc. You also have clients, routers, switches, as well as the wide variety of different systems that perform security tasks at different levels. This is usually referred to as an infrastructure. Most companies have this, and it's quite fascinating.

So, in this thing called infrastructure – you also have vulnerabilities. Either through bad design, implementation, wrong use or configurations of software at different levels or due to lack of maintenance. Some of these can be found and addressed quite quickly by the use of tools, while others needs manual testing before they reveal themselves.

I common approach is to do a full test using a lot of tools that address known vulnerabilities, common design flaws and such – in combination with penetration testing tools to sort of false positives and confirm what sort of consequences a breach would have. In combination with firewall policy analyzes, looking at the routines surrounding security all the way from development to maintenance you'll have some sort of baseline to work out from when it comes to the level of security. The work will also reveal how well the company can detect and address events.

So...Network Based IPS != Infrastructure.

When you have this baseline, and had time to address the issues you've found – you can start bringing this to the next level, which is what you bring in at the end of your response.

You need to know what you got first though, and for a lot of us – that is a huge task due to massive infrastructures (not Network Based IPS!).

But by doing this, you have examples of security flaws done by programmers, by people implementing solutions, by administrators and so on – and when you have this, and can point them out for people, it's easier to make them stop doing these mistakes.

Yeah – I answered on your trap, and I knew it would end up in another rant – like the ones you've been delivering the last 10+ years.

And yeah, I know that even though this looks like text to the rest of us, for you it's just a rorschach that makes you go off with a new rant – usually pretty far away from the subject.

Have a good one.

Re: Pentesting tool – Commercial

Andre Gironda wrote:

On Tue, Mar 4, 2008 at 4:47 AM, Trygve Aasheim <trygve@xxxxxxxxxxxxxx> wrote:

Trygve,

Thank you for walking into my strawman argument / trap.

So the results from your fuzzers can be implemented into modules in these tools and tested

You used the secret word! "Fuzzers"! Woohoo!

The framework can then run the exploits for you continuously while you test different configurations and version of the target software.

Exciting. I bet that will work wonders.

What type of weakness are you attempting to exploit in this example? Stack-based buffer overflows? While I understand the issues behind the need to verify the system state's properties per stack-based buffer overflow exploit (and this is getting rather complicated thanks to /GS, SSP, SafeSEH, ASLR, et al) – making a reliable overwrite on EIP (and/or several other registers) is far more complicated than these tools can even handle.

Not to mention that stack-based buffer overflows aren't the popular software weakness that they used to be. It's more difficult to find them and exploit them than ever before.

What about the other 642 software weaknesses?

Also it helps if you are looking at exploiting an infrastructure more than just running one exploit against one target

"Exploiting an infrastructure" is funny to me. I bet you are talking about network-based IPS. Or maybe host-based?

First question: What's the best thing about IPS?

<Awkwardly long pause>

Answer: NOTHING!

Second question: What's the best thing about pen-testing?

Re: Pentesting tool – Commercial

<Insanely long pause>

Like HD Moore and Valsmith's speech at Black Hat 2007, where they showed how to use the output from one module as input into another module – and then achieve your goals

Achieve your goals! Fight to win!

Btw – they're called `mixins'. It's a programming language thing.

(WPAD -> HASH -> login into Windows Domain example)

I knew about the problems with WPAD 10 years ago, and so did everyone I knew at the time. This sort of thing doesn't magically become possible thanks to Ruby.

The same approach are used by malware developers now, and mpack is a good example

What approach? You mean programming?

We're also seeing more and more fuzzingtools being implemented directly into these frameworks, like lorcon in Metasploit and the web attack modules in Impact

Lorcon is a driver. Web attack modules use fault-injection.

So then the tools can search for new vulnerabilities more than just act on the pre-loaded exploits.

I think we're going to have to redefine terminology if people really think this is true.

What does one do about the vulnerabilities that these tools don't find?

So I don't understand your limited view on these tools...  
It's like asking "if this car is so damn good, why can't it drive me to work...!?".

Re: Pentesting tool – Commercial

Lulz!@#! I don't understand your limited view on the vulnerability problem. Your opinion on pen–testing tools is like responding with "It's a floor wax; it's a cereal; it cleans your pipes!".

But that might be the difference here...you wanna exploit your iPhone, while these tools are made to test the security level of company infrastructures...

No, I don't want to exploit anything. I don't want anybody exploiting anything. I want people to find vulnerabilities and fix them, usually without tools. I want developers to instinctively avoid writing vulnerabilities thanks to some simple processes and workflow. I want formal specification and verification.

I don't want people to use penetration–testing tools to test the security level of their company. I don't care how secure that they are in this way. I want them to acquire secure software and use it securely. I want developers to write code securely.

How secure is Core Impact? Is it unbreakable? Can I write a listener that gets scanned and launches a shell through the scanner? Can I take control of the pseudo–worm?

Cheers,  
Andre

---

This list is sponsored by: Cenzic

Need to secure your web apps NOW?  
Cenzic finds more, "real" vulnerabilities fast.  
Click to try it, buy it or download a solution FREE today!

<http://www.cenzic.com/downloads>

---