Roelof Temmingh is the Technical Director of SensePost where his primary function is that of external penetration specialist. Roelof is internationally recognized for his skills in the assessment of web servers. He has written various pieces of PERL code as proof of concept for known vulnerabilities, and coded the world-first anti-IDS web proxy “Pudding”. He has spoken at many International Conferences and in the past year alone has been a keynote speaker at SummerCon (Holland) and a speaker at The Black Hat Briefings. Roelof drinks tea and smokes Camels.

Haroon Meer is currently SensePost’s Director of Development (and coffee drinking). He specializes in the research and development of new tools and techniques for network penetration and has released several tools, utilities and white-papers to the security community. He has been a guest speaker at many Security forums including the Black Hat Briefings. Haroon doesn’t drink tea or smoke camels.

Charl van der Walt is a founder member of SensePost. He studied Computer Science at UNISA, Mathematics at the University of Heidelberg in Germany and has a Diploma in Information Security from the Rand Afrikaans University. He is an accredited BS7799 Lead Auditor with the British Institute of Standards in London. Charl has a number of years experience in Information Security and has been involved in a number of prestigious security projects in Africa, Asia and Europe. He is a regular speaker at seminars and conferences nationwide and is regularly published on internationally recognized forums like SecurityFocus. Charl has a dog called Fish.

Automation - Deus ex Machina or Rube Goldberg Machine?

How far can automation be taken? How much intelligence can be embodied in code? How generic can automated IT security assessment tools really be? This presentation will attempt to show which areas of attacks lend themselves to automation and which aspects should best be left for manual human inspection and analyses.

SensePost will provide the audience a glimpse of BiDiBLAH - an attempt to automate a focussed yet comprehensive assessment. The tool provides automation for:

- Finding networks and targets
- Fingerprinting targets
- Discovering known vulnerabilities on the targets
- Exploiting the vulnerabilities found
- Reporting
Introduction

SensePost has done hundreds of external assessment
Tried and trusted methodology
So... in search of an automated assessment tool

This talk is about:
• What is this methodology?
• Can it be automated?
• Where does automation really work well?
• Where does it simply sucks?
• Why does it fail? (and can it be corrected?)
• Implications for penetration testers

digital self defense
Principles of automation

To have an automatic process we need to code it
To code it we need to have an algorithm or flow
In order to have an algorithm or flow we need to understand the process
To understand the process we need to have done it many times

If you cannot write the process down on paper you probably don’t understand it completely

Exceptions on the rule – the root of all evil

Tradeoffs – if it will work in 99.99% of cases and will take me 2 months to code support for the 0.01% of cases…is it worth it?

Weird perceptions

Unix good….Windows baaaad! (meeaaaaa)

‘Hard core’ hackers will tell you that Windows sucks.
GUI apps limit you to do complex things
Problem is not the OS – it’s the implementation of the GUI
People think that, because it’s a GUI app, it needs to be “dumbed down”
People think that, because it’s a GUI app, it needs to be user friendly
People think that, because it’s a GUI app, stupid people will use it

Unix tools are mostly “fire and forget”
Unix tools have difficulty showing progress
Unix makes it hard to write X11 interfaces – so ppl stick to text based interfaces
BiDiBLAH uses “hot” text boxes – you can copy and paste & grep and awk and sed all you wish

digital self defense
The demos you are about to see...

BiDiBLAH is a tool for doing attacks/assessments
   Its built for large networks
   ...we don't have a large network
   ...but our clients do
   ...but we don't want to show their network
   ...no...we don't...really...

SO:

Passive: IBM, Playboy
Active: SensePost/VMWare

There's just too much risk in doing this live
   ...but everything you see is real
   (some time lapse in places – I'll tell you where)

SensePost external methodology

Footprinting → Finger printing → Targeting → Vulnerability discovery → Penetration testing
Methodology: Footprinting

Find domains → Find sub-domains → Find forward DNS entries → Find netblocks/Define netblocks → Find reverse DNS entries → Perform vitality testing

Methodology: Footprint: Find domains

Initial domain → TLD expansion → Name expansion → Related domains

Content matching → Network (UDP/UDP) matching → Meta data matching → Final domain list
Methodology: Footprinting: Find subdomains

- Domain
  - Contains main domain?
  - Subdomains
    - Google keywords
    - Google Fu

Video – BiDiBLAH’s footprinting: Sub domains

digital self defense
Methodology: Footprinting: Forward DNS entries

- Domain / subdomain
- MX/NS records
- ZT possible
- Hit lists
- Perform forward
- All forwards

Video – BiDiBLAH’s footprinting: Forwards

digital self defense
Methodology: Footprint: Netblocks

- Trim forwards to class C
- Find the AS
- Find other routes in the same AS
- Clean netblocks
- Inspect reverse entries

Additional blocks:
- ARIN
- RIPE
- APNIC
- LACNIC
- AFRINIC

Video – BiDiBLAH footprinting: NetBlocks
Methodology: Footprint: Reverse DNS

Netblocks
- Perform reverse for each IP in block
- Filters
  - Match filter?
    - yes
    - More related domains
  - Unmatched entries
    - Extract domain
    - More related domains
- Matched entries
  - Extract domain
  - Do we have the domain?
    - yes
    - More name expanded domains
    - no
    - Unmatched entries
      - Extract domain
      - Do we have the domain?
        - yes
        - More name expanded domains
        - no

Methodology: Footprint: Vitality

Netblocks

Port hitlist

TCP scans

UDP scans

ICMP scans

IP list

Vitality: Async scanning

Sniffer

Flags & SRC port

SYN ACK

Send SYNs

Open ports

Send FIN

Closed ports

digital self defense
Video - BiDiBLAH – Vitality (SensePost network)

Initial domain
TLD expansion
Name expansion
Related domains
Network (MX/NS/IP)
Matching
Content matching
Meta data matching
Final domain list
Contains main domain?
Google keywords
Google Fu
Subdomains
Hit lists
ZT possible?
MX/NS records
All forwards
Perform forward yes
Trim forwards to class C
Find the AS
Find other routes in the same AS
Whois
Additional blocks:
ARIN
RIPE
APNIC
LACNIC
AFRINIC
Inspect reverse entries
Clean netblocks
Filters
Perform reverse for each IP in block
Match filter?
Matched entries
Extract domain
Do we have the domain?
Unmatched entries
Extract domain
More name expanded domains
More related domains
yes
no
TCP scans
UDP scans
ICMP scans
Port hitlist
IP list
digital self defense
Footprinting – a different view

Pheeww…glad that’s over!

Which steps are difficult to automate & why?

• Domain finding
  • works semi OK, but never complete [not implemented]
  • currently, you can learn a lot from reverse entries
• Sub domain finding – easy - [DONE]
• Forwards – easy - [DONE]
• Netblocks – difficult…
  • AS expansion is not always good for smaller (hosted) blocks.
  • Whois info on these blocks are pretty unless.
  • No standard interface to registrars
  • [Currently set to manual]
• Reverse scans – easy - [DONE]
• Vitality – easy [DONE (tcp only)]

digital self defense
So, where are we now?

Methodology: Fingerprinting

OS detection from the Internet to a firewalled host is difficult... Not just technically, but conceptually:
An Apache box protected by a FireWall-1 running on Win32 and 1:1NAT will report itself as a Windows machine on a network level... but as a Unix machine on app level... so what will it be??

BiD/BLAH does not try to do OS detection, but rather just do banner grabbing

Using Async banner grabbing for 21,22,25,80,110,143
Multithreaded 443 (SSL)
Any banner/version can be grabbed asynchronously but it gets increasingly tricky...
Async banner grabbing – the process

- Sniffer

  Is SYN_ACK?
  - Send ACK
  - Is ACK?
    - Extract banner
    - Port 80?
      - Send FIN/ACK
      - Is FIN?
        - Send FINACK
        - Port 80?
          - Extract banner from HTTP header
          - Port 80?
            - Send GET/HTTP/1.0
            - Send SYN

- Video - BiDiBLAH: Async banner grabbing
Methodology: targeting

With a great deal of potential targets, we want to be able to select only those that really interests us.

Targetting system should be able to target using
- Certain/All open ports (in all netblocks, or certain netblocks)
  - e.g. all open on TCP 53
- Keywords in service banners
  - e.g. wuftp*
- Keywords in DNS names
  - e.g. PRT*
- All hosts in a specific netblock
  - e.g. all in 172.16.43.0/24
- Particular OSes of version of OS [a problem - we don’t have it]
  - e.g. MS Windows XP SP1
- Certain keywords within vulnerability descriptions (later more)
  - e.g. RPC*
Video – BiDiBLAH - Targeting

SensePost external methodology

So, where are we now?

Footprinting → Finger printing → Targeting → Vulnerability discovery → Penetration testing

digital self defense
Methodology: Vulnerability discovery

Why reinvent the wheel? Use a solid, widely used scanner: Nessus...

Thus...we write a Nessus client.
Give the user the ability to choose a set of plugins
..and let him save the list..

Thus – you can choose "all" plugins (if you are doing an assessment), or you can choose one plugin (if you are looking throughout your whole network for a particular problem)

Scans are executed against what was marked as targets
So, where are we now?

- Footprinting
- Finger printing
- Targeting
- Vulnerability discovery
- Penetration testing
Methodology: Vulnerability exploitation

Why reinvent the wheel? Use a solid, widely used exploitation framework: MetaSploit!

Thus…we write a MetaSploit client..

Problem with MetaSploit – its very operating system specific ….and we DON’T HAVE that…

Don’t specify target and hope for the best – hopefully it will brute force.

Use Nessus to identify the weakness, MetaSploit to exploit it
Thus … we need a NessusID to MetaSploit sploit name list
We built it (thanks GP), and wrote plugins as needed
Hopefully it can be an attribute of the sploit (looks at HD..)

RHOST, SSL, LHOST – all known to us
RPORT known via Nessus scanner
Let the user choose the payload and additional parameters

digital self defense
SensePost external methodology

Footprinting ➔ Finger printing ➔ Targeting ➔ Vulnerability discovery ➔ Penetration testing

So... we are done?

In a perfect world... yes...

In the real world we have false positives, we have to moderate Nessus results, and we have to write

!="|||ing reports!!!
The Bottom line

BiDiBLAH does 80% of the work within 20% of time it takes us
The last 20% of the work takes 80% of the project time

Some steps in the methodology are really hard to automate
This is usually where things are “non-standard”, or an exception

It would hopefully raise the bar on mediocre “pen testing” companies

Release considerations

Group1: “Surely you will not release this to the world – you arming
script kiddies with dangerous point and click hacking tools!!?”

Group2: “Where do we download it? What you mean pay for it?! You
just pushing product here!!

Thus: crippled version released at
http://www.sensepost.com/research/bidiblah
Commercial version available on request

EXTRA: E-Or release

Web APPLICATION assessment tool
•http://www.sensepost.com/research/eor