Rain forest puppy

Hack/03 - Australia

10.9.200.197

RFP's collection of random misconceptions

10.9.200.197

Misconception:

The annual pen-test is sufficient

- Difference between a pen-test and an assessment
- “What can the ‘hackers’ do…”
- Quality of test linked to skills (and 0day) of who you hire
- Imposed limitations hamper “hacker perspective”
- Typically don’t allow exploit avenues available to attackers
- Proves insecurity; does not prove security

Effective uses of pen-tests:

- Stagger different firms and compare results
- Use cheaper pen-tests to prove insecurity in order to get management buy-in for more expensive assessments/audits
- Verify your incident response team is active
- If you really want to gauge threat, don't impose any limitations

Misconception:

We can solve the security problem by throwing products at it

- We’ve become a product-laden industry:
    - Anti-virus (Symantec)
    - Firewall (Checkpoint)
    - IDS (Snort)
    - IPS (TippingPoint)
    - [SSL] VPN (Neoteris)
    - Vulnerability assessment (Nessus)
    - Web content control proxy (SurfControl)
    - SIMs (Arcsight)

- Products continued…
    - Web application firewall (Interdo)
    - XML firewall (Datapower)
    - Development review tools (AppScan DE)
    - Specialty scanners (WebInspect)
    - Patch management (Shavlik)
    - Honeypots (Mantrap)

- It has become the “quest for the perfect product”
- Very few products cover more than one area
- Shortcomings often lead to ‘gaps’ in coverage, which may require a second product to help complete coverage
- Tossing products at security is a sure-fire way to squander your budget
- Management of all these devices becomes a nightmare
- Vendors may over-hype/misrepresent features

Misconception:

Just put a firewall in front of it

- Traditional use of layer 2/3 firewall is obvious
- Even with a firewall, port 80 can still be exploited
- Recent surge of application-layer firewalls:
  - Generic protocol proxies (included with FW1, Gauntlet, etc)
  - HTTP inspection firewalls (a.k.a. webapp firewalls, web security proxies)
  - XML/web services firewalls

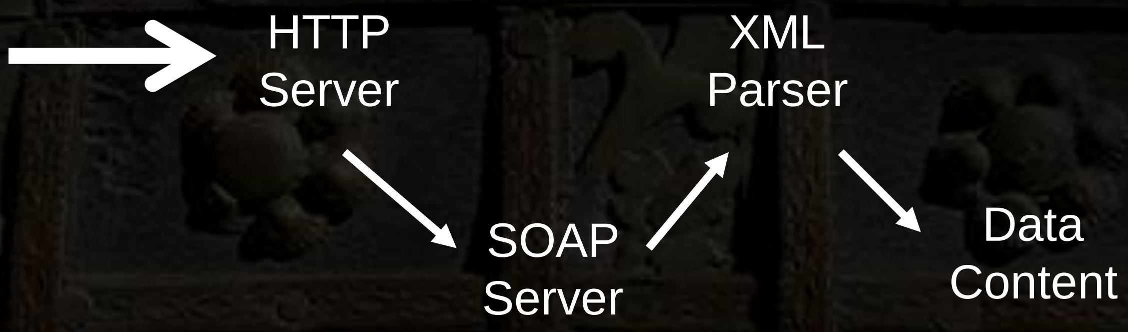
Focus: XML firewalls

- Used to protect SOAP and other XML services
- Many offload XML and schema validation
- Some handle WS-Security extensions

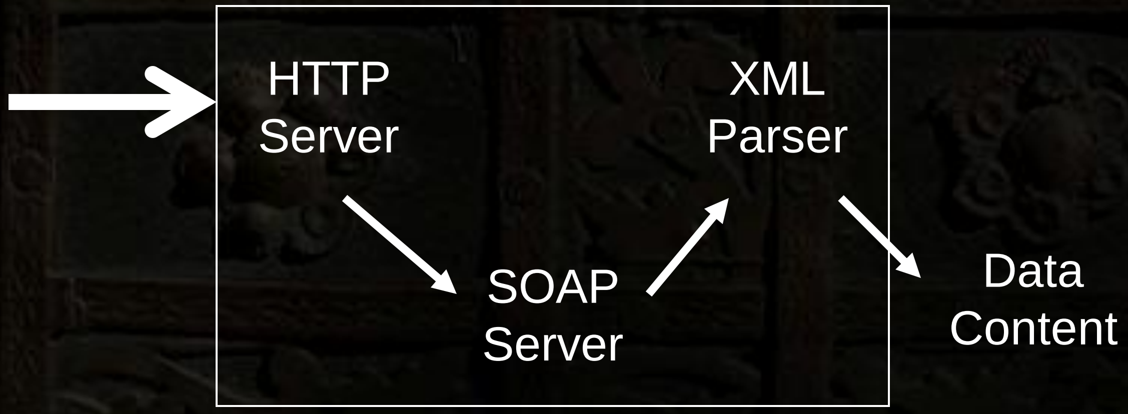
Largest problems with SOAP:

- Access control on granular level
- Lots of processing layers = lots of room for bugs
- Data level bugs (SQL tampering, buffer overflows, etc) still present

Standard SOAP process flow



XML Firewall



Largest problems with XML firewalls:

- Some security validation defined by the WSDL, but many dev tools won’t generate security-conscience WSDL definitions
- Little to no application content validation, or mechanism to insert your own filtering
- No support to provide security for non-SOAP (regular HTTP) content
- No support for WebDAV

Focus: Webapp firewalls

- Used to protect HTTP server, CGIs, web apps
- Inspect and validate every request
- Great way to ‘lock down’ applications which you don’t/can’t control
- Come in network/proxy and host plugin models

Largest problems with Webapp firewalls:

- Configuration required for every text box on every form on every page on every web server…
- You have to be a regex guru
- And often you have to be a web hacking guru
- No support for WebDAV
- No support for application-level or tunneled protocols over HTTP: Microsoft FrontPage, Java RMI, RTSP, etc.
- Typically require a layer 2/3 firewall in front

Regex guru example:

I want to only allow GIFs

```
/.*\.gif
        /../../../../passwordfile.txt%00.gif

/[a-z0-9]+\.gif
        /yousuck.gif/../../../../../passwordfile.txt

^/[a-z0-9]+\.gif$
        /unexploitable.gif
```

Time for some interactive demo!

10.9.200.197

Imagine…

You are a ruthless hacker who has just wandered across a vulnerable Windows NT 4.0 system running an insecure version of IIS 4.0.

This system has *tons* of confirmed vulnerabilities. Uber-hacker or script kid, everyone should have something they can exploit.

Time to impress your friends by breaking into it!

10.9.200.197

The catch…

It's protected by two different web application firewalls.

In order to exploit any of the bugs, you'll need to find a way around the webapp firewall(s), or perhaps just find an oversight or misconfiguration…

10.9.200.197

The challenge…

Log onto wireless and start poking at the demo webserver:

IP address: 10.9.200.197

Port 80: Kavado Interdo
Port 81: Sanctum AppShield

10.9.200.197

The rules…

- No DoS attacks (what’s the point…)
- Wireless/network attacks are irrelevant
- No need to port scan: only port 80 and 81 open
- Make use of the AppScan report to get started
- Some tools and exploits you may need are downloadable from the site itself

10.9.200.197

Walkthrough…

10.9.200.197

Break time

Now back to your regularly scheduled misconceptions…

10.9.200.197

Misconception:

PC virtualization is the way to go in production environments

- Virtualization = use of VMWare to run multiple virtual production systems on one physical system
- Stems from the mainframe mentality

Wacko justification #1:

“If we move it all onto one box, then that makes patching easier”

Reality: you now have N+1 systems to patch, and patching is not any more consolidated; don’t forget the VMWare patches too (which are now critical)

Wacko justification #2:

“We can reduce hardware support contract costs”

Reality: If the hardware is so reliable, then why do you have support contracts? If the hardware is *not* reliable, then why are you moving it all onto one box?

Wacko justification #3:

“This model worked well for mainframes”

Reality: mainframes were designed for virtualization; PCs were not

Good uses for virtualization:

- Research, testing
- Staging
- Development

Systems/environments that are only occasionally needed work well with virtualization

Product system virtualization = all your eggs in one basket

Misconception:

We'll perform a 'wireless audit' to find rogue wireless devices

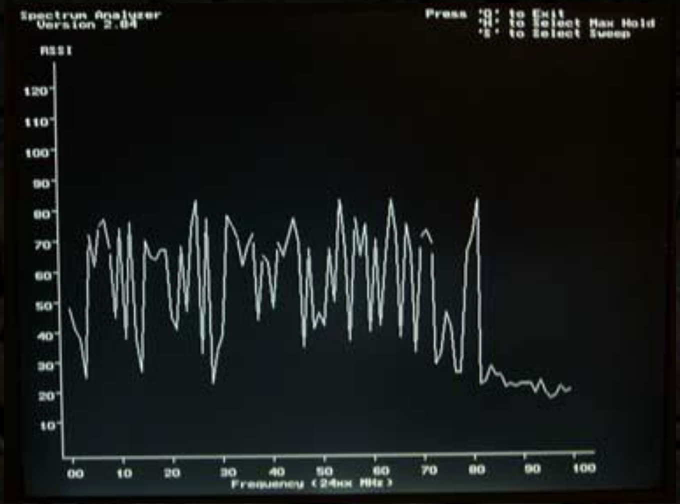
- Desire to find any rogue wireless devices
- Concerns about wireless entry into network
- However, tend to only look for popular protocols:
    - 802.11b, maybe 802.11a or 802.11g
    - Bluetooth (?)

- No one looks for full suite of protocols:
  - 802.11 FHSS
  - 802.11 & 802.11b (DSSS)
  - 802.11b-turbo (vendor specific)
  - 802.11a
  - 802.11g
  - HomeRF
  - Bluetooth
  - Vendor-specific (e.g. WebGear Aviator)
  - Cellular & CDPD

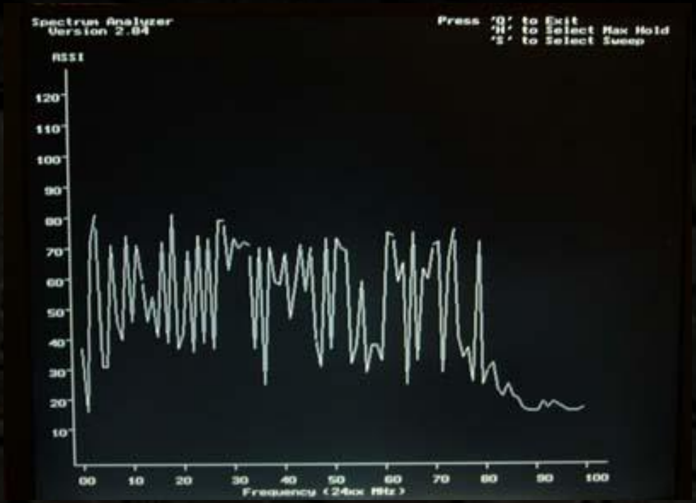
- Not to mention:
    - HomePNA
    - Powerline networking
    - X10 wireless cameras
    - Wireless keyboards
    - Infrared
    - SMS
    - Blackberry and other do-hickeys

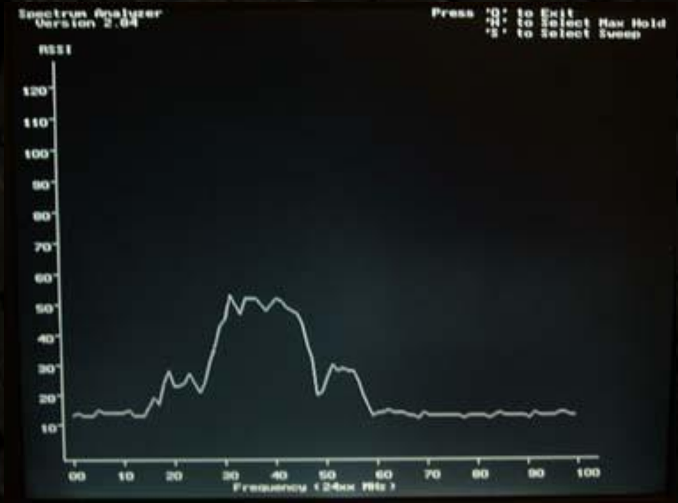
- Lots of equipment is required to support/search for so many protocols
- There are WLAN detectors
- A more concrete solution involves a spectrum analyzer…

802.11 FHSS

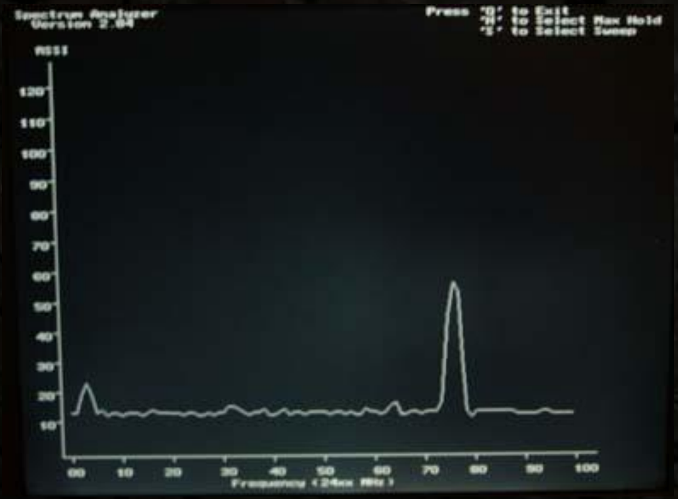


HomeRF





802.11b DSSS



GE Cordless Phone

Solution?

• Auditing is tedious and painful at best

• Many things are likely to go undiscovered

The ultimate solution: Faraday cage?

(hey, it works for the US government…)

Misconception:

Wireless authentication is not that important

- Need a way to prevent unauthorized WLAN use; typically means you need to authenticate users before allowing wireless access
- Wireless is sniffable
- WEP is crackable

Result: how do you authenticate a user in a manner which doesn’t give away their username/password?

BTW, don’t even think about tokens…

Quasi-fix #1: fix the encryption

- WPA: WEP replacement
- 802.11i, eventually (WPA and RSN)

However, auth is separated from encryption, and may occure *before* encryption is established

Uber-bad idea: authenticate to your domain

- Recovery of user auth info via wireless now allows attacker access to other enterprise systems
- Password lockout can lead to DoS problems

Popular wireless/EAP authentication protocols:

- EAP-MD5: send an MD5 password hash
- EAP-TLS: client and server certs
- EAP-TTLS: server certs, plaintext password
- EAP-Cisco (a.k.a. LEAP): NTLM challenge/ response

Focus: Cisco LEAP

- EAP auth happens before encryption/TKIP
- You can sniff the challenge and response

Result: offline dictionary attack

Cisco security advisory 20030802-leap

Further, same challenge/response is used in MS PPTP; Counterpane and L0pht published shortcuts to cracking

Misconception:

We wait XX time before applying Microsoft patches

- Standard response is 1-2 months
- That’s what staging environments are for…
- It’s easy for 1-2 months to turn into 6-12 months
  e.g. Code Red
- If you don’t trust a vendor, why do you run their stuff?
- Worm generation has reached <30 days

Misconception:

IDS is dead; long live IPS

• IDS is shunning traffic willy-nilly

• Attack detection based on current IDS technology

• Who hasn't received a false positive from an IDS?

Enough said.

Large potential for DoS and business interruption

Let’s check in on the demo server…

10.9.200.197

Anyone break in?

10.9.200.197

Disclaimer…

The purpose of this exercise was to give credibility to webapp firewalls and their ability to protect (very) insecure hosts.

It was not meant to demonstrate or convey the idea that webapp firewalls are a substitute to patching.

10.9.200.197

Closing thoughts:

Misconceptions about hackers

- Differentiation between script kiddie and uber hacker
- How do attackers ‘operate’?
- What kinds of stealth are they using nowadays?
- What are they targetting?
- How fast do they attack?
- What new tricks are they using?

Questions

&

Thanks.

10.9.200.197