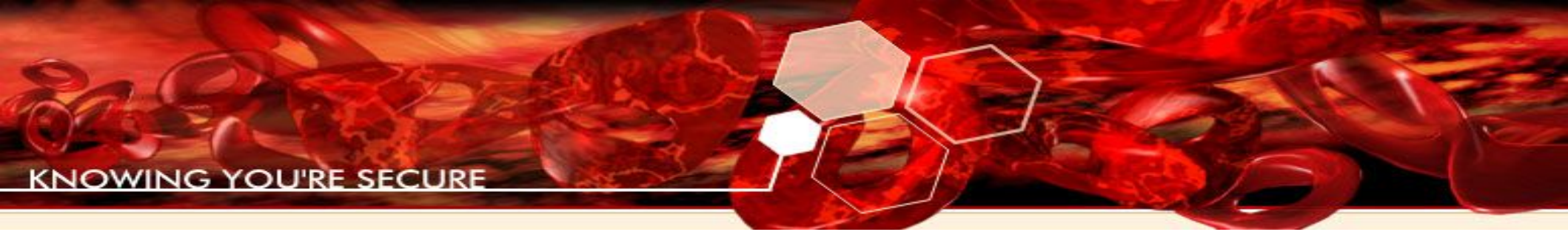


0days: How hacking really works

V 1.0
Jan 29, 2005

Dave Aitel
dave@immunitysec.com

Who am I?

- NSA->@stake->Immunity
- CEO of Immunity, Inc.
  - Consulting (product assessments)
  - Immunity CANVAS
  - Immunity Partner's Program
  - Training
  - Ongoing research in exploits and software vulnerabilities

Common Questions

- Why have I been hacked?
- How have I been hacked?
  - Specifically
  - Generally, how could this happen to me when I put all that money into firewalls and patching systems?

Agenda

- Examine different types of vulnerabilities from a hacker's standpoint
- Look at the future of hacking
- Look into the future of defensive measures

Quick note

- Some of the following slides are from a hacker's perspective
- We're not backing this up with academic papers and equations, consider it all opinion

Exploits vs Vulnerabilities

- An exploit is a working program that takes advantage of one **or more** vulnerabilities in order to break security boundaries
  - A good exploit often costs a lot more to develop than the initial cost of discovering a vulnerability
- A vulnerability may be something as simple as a memory leak or DoS
- It's hard to say if a vulnerability is exploitable without an exploit
  - GOBBLES and Apache

Working Exploits

- What does a hacker want to know about a given exploit?
  - Reliability
    - “Will this work in the wild?”
  - Target set
    - “Do I even care if it does?”

Exploit Reliability/Usage

- Logging
  - Logging can be both too succinct to be useful, or two verbose
- Does the service restart vs. One-Shot
  - Many Windows services are one-shot attacks, but Win32 threading models can make for very reliable one-shot attacks
- Failure modes
  - Even very good exploits fail sometimes

Target Set

- Interesting boxes?
  - SSHD vs SADMIND vs WUFTPD
- Default/common configuration?
- Multiple configurations?
  - Increase in targeting complexity
- Is this an exploit I can easily scan for?
  - fingerprinting

Survivability

- Exploits require large amounts of investment
  - Scanning/fingerprinting is non-trivial
  - QA on a complex piece of software is expensive
- How long is this vulnerability going to be valid?
  - Turn “windows of vulnerability” upside down
  - Multiple independent discoveries are more the rule than the exception

Easy vs. Hard Targets

RealServer ⟷ dtlogin

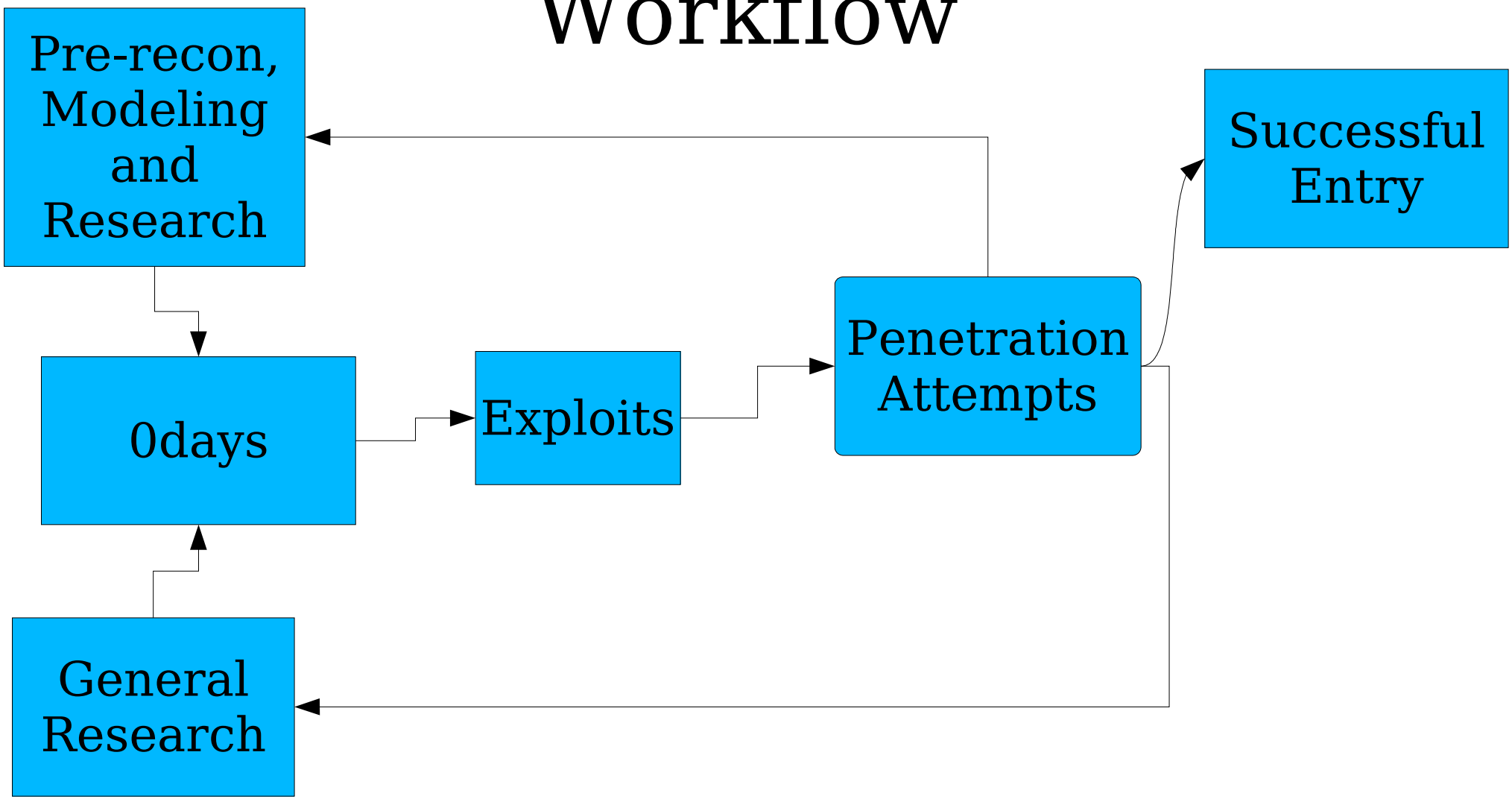
Where do I invest my time?

- Realserver: Multi-shot target-less self-fingerprinting stack overflow
- dtlogin: one shot heap corruption
- Compounded by question: What are your most important targets running?

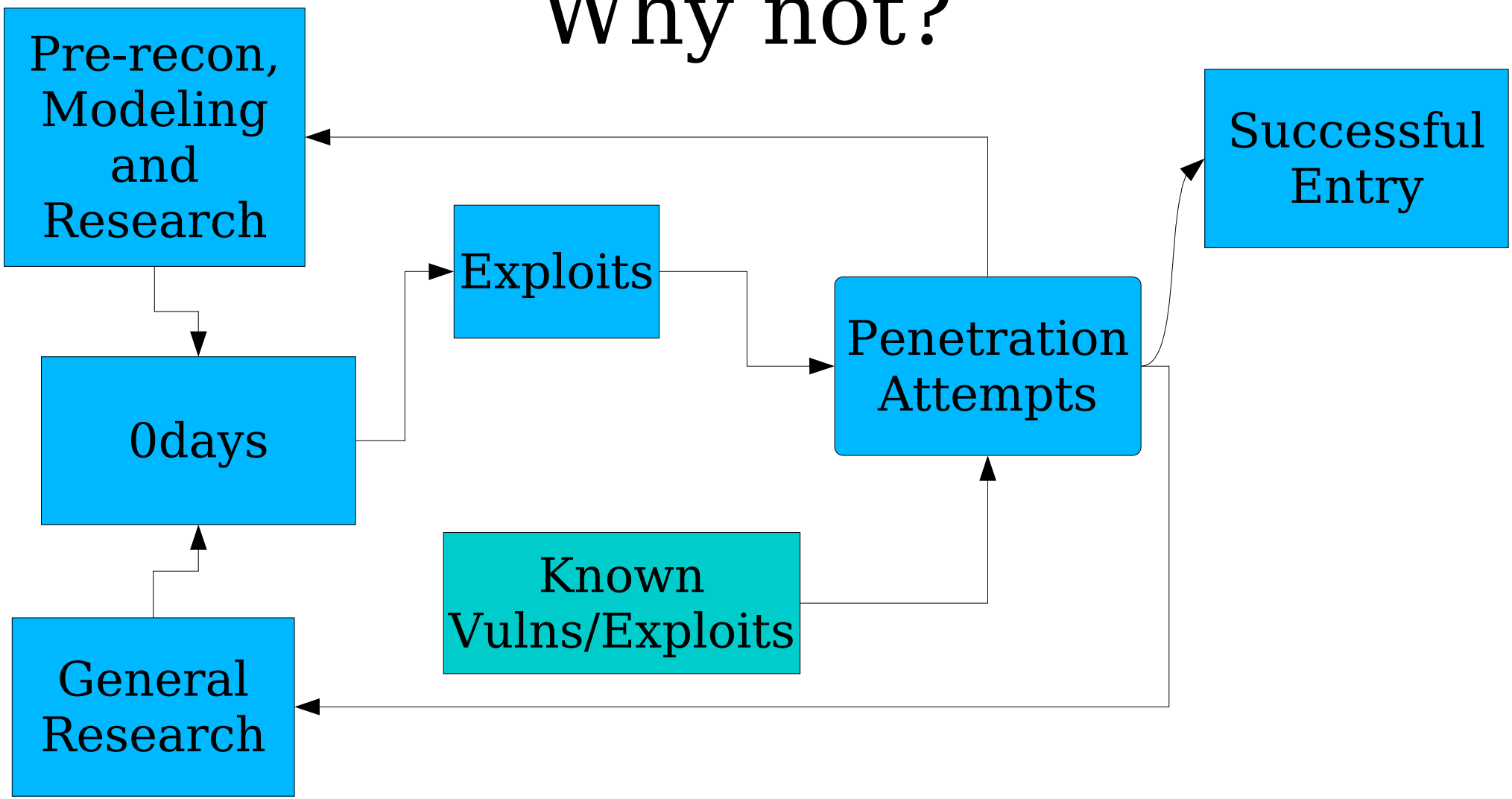
Custom Exploitation

- The most covert exploit is one that is used only once
  - Custom web application hacking
  - Custom analysis of target's environment
    - Example: Exploit for cam.exe with Entercept installed in the exact configuration you have for all your servers

Workflow


Pre-recon, Modeling and Research
Successful Entry
Penetration Attempts
0days
Exploits
General Research

Why not?


Pre-recon, Modeling and Research
Successful Entry
Exploits
Penetration Attempts
0days
Known Vulns/Exploits
General Research

Why not to use known vulns/exploits

- A bad investment, even if it works
  - May be detected by IDS, allowing target to track your methodology and toolkit
    - toolkits are expensive ($100K->$1M)
    - methodologies are more expensive
      - a trained team: $1M->$10M
- Worse, if it doesn't work
  - Each attack burns a bounce host
  - Each attack alerts target they are under attack

One shot, one kill

- But we have to make all our bullets by hand
  - Is it logistically possible to write an 0day for each target?
  - What is the “cost” of using an exploit?
- Our toolkits and methodologies are even more expensive
  - Can we afford complete duplication of effort?

Writing an 0day per target network

- Costs
  - Between $10-100K per network for a given exploit
- Benefits
  - Research can be version specific (cuts costs)
  - No IDS catches you
  - Getting caught does not blow other targets
    - backwards operational security is as valuable as forwards

Windows of Vulnerability

- Arbaugh, et. al. in 2000 IEEE Computer paper
  - http://www.cs.umd.edu/~waa/vulnerability.html
  - (2002) Active Systems Management: The Evolution of Firewalls
- Accepted general model of security industry
  - To defeat the industry, hackers have defeated this generic model

Intuitive?

IEEE (Dec 2000) Arbaugh Fithen, McHugh

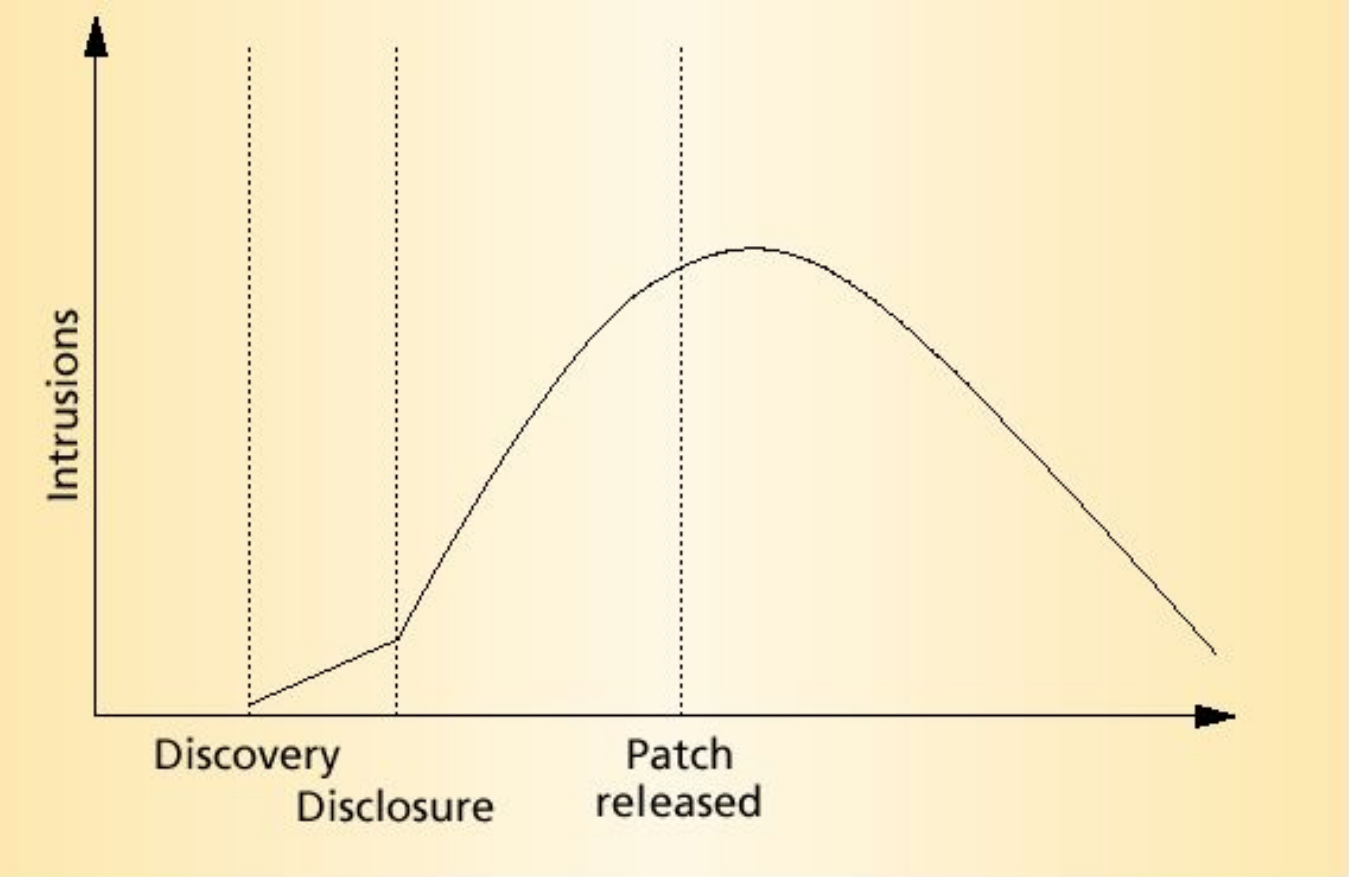
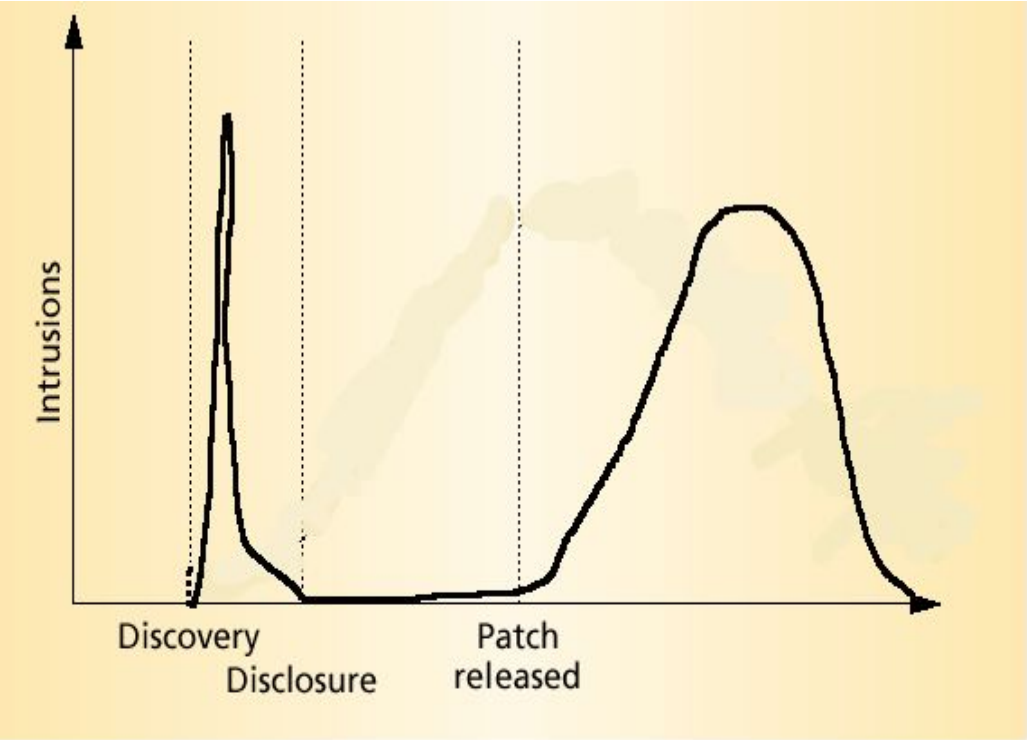


Figure 1. Intuitive life cycle of a system-security vulnerability. Intrusions increase once users discover a vulnerability, and the rate continues to increase until the system administrator releases a patch or workaround.

Hacking is not theoretical

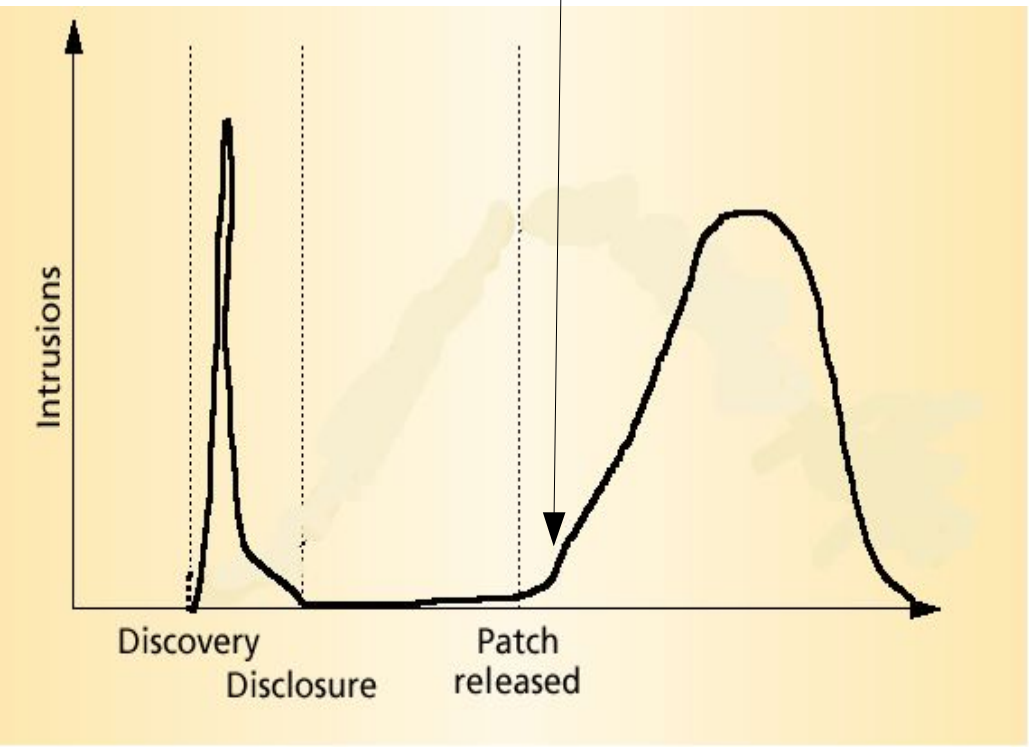
- Hackers do not own every machine that has a given vulnerability
  - that would be stupid
- Hackers own every important box that they do not already own
  - Generic and specific automation is as old as exploits
    - Admhack, etc.
- It is fair to say hackers have a generational lead on the industry

A closer curve



- Most interesting machines are owned shortly after discovery. Discovery rarely happens by “researchers” first.
- Patch information releases a lot of information about the vulnerability.
- Upon disclosure, real hacking stops.
- Hackers have access to a lot more “Internet” than the average public or a worm
  - Most vulnerable machines are on intra-nets

Why doesn't my IDS report this?



IDS becomes potentially effective here.

- Entire study is based on **<u>discovered</u>** intrusions!
  - (vs. attempted intrusions?)
- Are we measuring detections, rather than intrusions?

Passwords

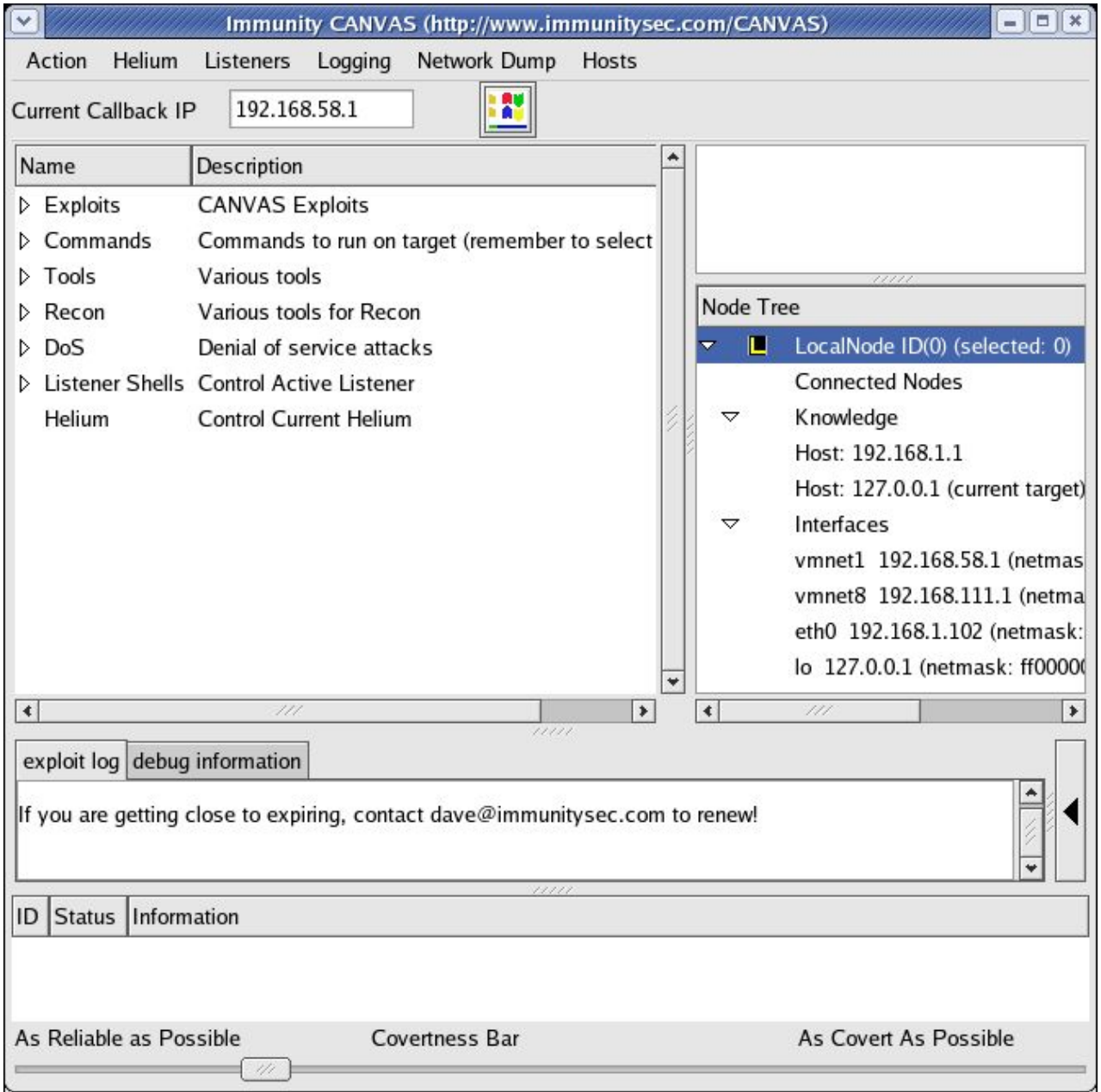
- Are still the best way to protect information systems
  - great manageability interoperability, etc
- Are also the best way to hack into systems
  - known_hosts
  - password reuse is universal

Hacker Network Targets

- Nervous systems are the primary target
  - Management networks
  - Intrusion detection networks
- Software companies add to this:
  - Security departments and QA systems
    - Predicted: A small bounce when vulnerabilities are reported

Looking towards the future of Attack

- More automated frameworks, public and private
- 0day and more 0day
- Customized worms



Automated Attack Frameworks

Public, high quality rootkits

VMWare

Reduced vulnerability disclosure

0day and more 0day

- As systems get more protected 0day becomes more valuable
- Survivability of even very popular 0day is measured in years, if not decades
  - Sadmind
  - VSC results
- Web application vulnerabilities are just the beginning

Customized Worms

- Custom worm generation languages
  - AdvancedOrdnance
  - Automated frameworks ARE worms (hydras)
- Worms are also useful for enterprises looking for distributed techniques
  - Think of them as distributed computing writ large
  - Write applications with worms as your platform

Looking towards the future of Defense

- The failure of patching
- Universal Configurations (automated patching)
- HIDS
- OS Protection

Patching is basically useless for security

- You must reinstall all vulnerable systems, reset all passwords for security
- This is an unattainable goal
- Patching quickly is extremely expensive
  - manpower, resources
  - mistakes are costly
  - still not winning race

Universal Configurations

- Mono-cultures are a known evil
- Management software is typically weakly secured
  - Computer Associates cam.exe, Naimas32,etc
- Custom exploits are best against universal configurations
  - From custom exploits to custom worms

HIDS

- HIDS products receive little 3rd party testing
- Phrack 62 describes some widely known techniques for bypassing common HIDS technology
- You need a HIDS that prevents attacks, not shellcode
- HIDS are too expensive, by far

Network Intrusion Prevention Systems

- NIPS has a very very hard problem
  - Must model all types of systems and protocols
  - Must correctly detect attacks while in stream to target
  - Must know about all different variants on attacks
  - This is all exponentially expensive stuff
- Good against worms

OS protection

- Windows XP SP2
  - Should be required
  - Not perfect
    - Immunity has generic techniques to bypass it, so assume hackers do as well
  - IE is impossible to truly secure, ban it if possible
- Linux is much better (GRSecurity)
- Unix is much worse

Regulation

- No presentation is valid these days without a slide on Sarbanes-Oxley
  - This is that slide

Conclusion

- Use GRSecurity or HIDS
- Don't rely on patching as a security measure
- Get third party reviews of critical custom software
- Your intrusion response team is only really tested by 0days
- Stop purchasing junk software and then blaming other people for your problems

Questions?

- Did we answer more than we asked?