Why Black Hats Always Win

Val Smith (valsmith@attackresearch.com)
Chris (chris@sdnaconsulting.com)
Val Smith

– Affiliations:
  • Attack Research
  • Metasploit

– Work:
  • Attack Techniques Research
  • Pen Tester/ Exploit developer
  • Reverse Engineer
  • Malware Analyst

Previous Talks
– Exploiting malware & vm detection
– Kernel mode de-obfuscation of malware
– Data mining malware collections
– Tactical Exploitation
– Post Exploitation
– Analysis of foreign web attacks
Chris is a Security Consultant and Researcher with Secure DNA. Chris specializes in web based application development security. He has collaborated with some of the top security researchers and companies in the world and has performed static and dynamic security assessments for numerous companies and government agencies across the U.S. and Asia.
What are we talking about?

• Overview of:
  – White hat Methodologies
  – Black Hat Methodologies

• Attackers VS. Defenders

• Analysis of Black Hat techniques in the Wild

• Black Hat Methodologies Demystified

• How can this help you?

• What can you do?
Overview of White Hat Methodologies
Overview of White Hat Methodologies

• **Goals**
  – Focus on racking up numbers of hacked machines
  – Data to fill reports
  – Identifying mitigations
    • How to prevent the attack
  – Vulnerability footprint, not penetration
    • Often identifying accessible data is secondary goal
Overview of White Hat Methodologies

• Goals
  – No downtime for the customer
    • DoS usually not allowed
    • Even if it facilitates access via reboot, etc.
  – No modifications
    • Typically can’t change:
      – Customer source code
      – Databases
  – Testing the response and detection mechanisms
    • Did the IDS catch us? Did they do anything?
Overview of White Hat Methodologies

• Information Gathering
  – Heavy focus on scans
    • Massive NMAPs / Nessus normal
  – Some overlap with Black Hat's
    • DNS / Domain lookup records
    • Google hacking
    • Personnel googling
  – Less concern for detection
Overview of White Hat Methodologies

• Vulnerability Assessment
  – Almost always automated scanners
    • Detectable & fingerprintable
  – Often a guess at potential vulnerability
  – Focus on risk & threat analysis
    • Vulnerability Consequences
      – How does this hurt client business
      – Do they stand to lose money / customers?
      – How likely is attack to occur
Overview of White Hat Methodologies

• **Exploitation**
  - Download and run exploits from milworm
    - Now defunct
    - How many pen test shops does this put out of business?
  - Securiteam & Security Focus
  - Core Impact / Canvas / Metasploit
  - Match up with nessus results
  - Usually no testing, run live against customer
Overview of White Hat Methodologies

• **Data Collection**
  – Screenshots
  – Sample documents
    • Just enough to prove access
  – No Analysis of attack paths
  – No prolonged infiltration
    • No long term sniffing / keylogging
Overview of Black Hat Methodologies
Overview of Black Hat Methodologies

• Goals
  – Wide ranging
  – Data, not just access focused
  – Targeting specific trusts
    • People weakest link in trust chains
  – Semi-unrelated access that may provide stepping stone
    • 6 degrees of separation
    • Any box on any network 6 degrees away from true target
Overview of Black Hat Methodologies

• **Goals**
  – Access to source
    • Let THEM do the hacking for you
      – They infect their own systems with backdoored updates
    • Source enables more assets
  – Example:
    • Target runs wordpress
    • Black Hat owns wordpress source server
    • Audit & Backdoor code
    • Surefire ownage of ultimate target in time
Overview of Black Hat Methodologies

• **Information Gathering**
  – Nothing is off limits
  – If needed info resides on unrelated box its still in scope
  – Social networking
  – Call up target and ask for info
    • Call targets friends, co workers, family
Overview of Black Hat Methodologies

• **Vulnerability Assessment**
  – Attacker’s often know what’s vulnerable ahead of time
    • No need for noisy scans
  – More efficient method than white hat’s trial & error
  – Stolen source code
    • Trojaned
    • Audited for 0days
Overview of Black Hat Methodologies

• Vulnerability Assessment
  – Non-traditional vulnerabilities
  – Example:
    • Software distro & licensing application
    • In house written by target
    • Installed on every computer
    • Runs with domain admin account privileges
    • Password changed every x min time interval
      – Accessible clear text in memory with debugger
    • Domain admin access to any machine for x minutes
Overview of Black Hat Methodologies

- **Exploitation**
  - 0 Days
    - Often only used when public bugs don't work
    - Avoid risking burning unpublished bug if possible
  - Usually interception from another box is better
  - Ex. Metasploit usually waits for 0day to become public before trunking
  - Wait till bug becomes 1day then blend in with worm traffic
Overview of Black Hat Methodologies

• **Data Targets**
  - Mail spools
  - Backup files
  - Database dumps
  - Sniffer logs
  - Keystrokes and chat logs
  - Access tokens
    • Crypto keys, kerberos tickets, windows domain tokens
  - Targets of opportunity
    • Maybe data xyz is the goal but abc is found more valuable
Overview of Black Hat Methodologies

• Data Theft
  – Client Injection / Exploitation
    • Vulnerable Client Applications
      – BSD IRC client exploit
    • Browsers
      – Grab sensitive data in browser POST
        » Before its SSL encrypted on screen keyboards = useless
  – Backdoors
    • Access Points
    • Services
    • Utilities
Attackers vs. Defenders
Attackers vs. Defenders

- **Defenders:**
  - Limited resources
  - Limited time
  - Rules of engagement
  - Consequences based on performance
    - If a pen tester never gets in, they stop getting hired
  - Motivation

- **Attackers:**
  - Unlimited resources
  - Unlimited time
  - On a long enough timeline everything gets owned
  - If attacker targets you, odds of success increase over time
  - No consequences to not getting in
  - Little to no rules
  - Motivation
Attackers vs. Defenders

- White Hats usually assigned limited block of IP addresses
- Unable to go beyond the scope of approved list

- Black Hats usually know one piece of information and have to expand from there
  - Domain Name
  - Email address
Attackers vs. Defenders

• Black Hats need techniques for discovering target related IPs and client side info
  – News group mail header harvesting
  – Proxy log analysis site mining
  – Backscatter spam
  – Botsvsbrowsers
You know the target’s domain name
Look at the IP range
Unlikely to be the target’s operational LAN

Domain Name: SINA.COM
Registrar: NETWORK SOLUTIONS, LLC
Whois Server: whois.networksolutions.com
Name Server: NS1.SINA.COM
Name Server: NS2.SINA.COM
Name Server: NS3.SINA.COM
Name Server: NS4.SINA.COM
Name Server: NS5.SINA.COM
Name Server: NS6.SINA.COM
Name Server: NS7.SINA.COM
Name Server: NS8.SINA.COM
Name Server: NS9.SINA.COM
Name Server: NS10.SINA.COM

Domain Name: SINA.COM
Registrar: NETWORK SOLUTIONS, LLC
Whois Server: whois.networksolutions.com
Name Server: NS1.SINA.COM
Name Server: NS2.SINA.COM
Name Server: NS3.SINA.COM
Name Server: NS4.SINA.COM
Name Server: NS5.SINA.COM
Name Server: NS6.SINA.COM
Name Server: NS7.SINA.COM
Name Server: NS8.SINA.COM
Name Server: NS9.SINA.COM
Name Server: NS10.SINA.COM

>>> Last update of WHOIS database: Sat, 23 May 2009 23:57:42 UTC
Query WHOIS networksolutions.com with 'SINA.COM'

Domain Name: SINA.COM
Registrant Name: SINA.COM
Registrant Organization: SINA TECHNOLOGY CHINA CO., LTD
Registrant Address: BEIJING, IDEAL PARK, NO.58 NORTHWEST 4TH RING SOUTH JIADIAN STREET, BEIJING, CHINA
Contact Email: info@sina.com
Contact Phone: +86 10 5869 1000
Contact Fax: +86 10 5869 1001

Address lookup
Ccanonical name www.sina.com.cn
addresses 71.5.7.191
Searching newsgroup postings for the target domain yields an email bounce with headers.

Header shows the IP the email was sent from.

Likely to be the target LAN or a home IP of a user on the target LAN (VPN maybe?)

Sometimes the headers in mailing list posts themselves have the same info.
Check the IP the email came from

Totally different network, in the target country
Search for file types associated with mail boxes to gather client side information.
From stephan.jau@apandrews.com Sat Apr 26 17:38:24 2003
MBOX-Line: From stephan.jau@apandrews.com Sat Apr 26 01:28:41 2003
Message-Id: <5.2.0.9.2.20030426102623.025ce3b8@mail.spamcop.net>
X-Sender: stephan.jau@apandrews.com@mail.protgp.com
X-Mailer: Qualcomm Windows Eudora Version 5.2.0.9
In-Reply-To: <16040.65254.766438.720746@hyde.home.loc>
Mime-Version: 1.0
Content-Type: text/plain; charset="us-ascii"; format=flowed
X-Spam-Status: No, hits=-4.9 required=5.0 tests=IN_REP_TO,DEAR_SOMEBODY version=2.20
X-Spam-Level:
From: Stephan Jau <stephan.jau@apandrews.com>
To: pelissero AT tiscali DOT de
Subject: Re: pelissero.org
Date: Sat, 26 Apr 2003 10:28:38 +0200

Dear Walter,
Bots vs browsers gives you by IP address client information such as browser and operating system.
Some sites have exposed squid proxy log analysis pages.

In this view you can see some hostnames and internal IP addresses.
This view shows userIDs and traffic quantities.

### SARC Square Analysis Report Generator

**Squid User Access Reports**

**Period:** 2009May22-2009May22  
**Sort:** BYTES, reverse  
**Topuser**

<table>
<thead>
<tr>
<th>NUM</th>
<th>USERID</th>
<th>CONNECT</th>
<th>BYTES</th>
<th>%BYTES</th>
<th>IN-CACHE-OUT</th>
<th>ELAPSED TIME</th>
<th>MILISEC</th>
<th>%TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>adminhotel</td>
<td>13.09K</td>
<td>247.35M</td>
<td>31.30%</td>
<td>0.80%</td>
<td>99.20%</td>
<td>11:24:34</td>
<td>41,074,091</td>
</tr>
<tr>
<td>2</td>
<td>filippova</td>
<td>8.95K</td>
<td>156.79M</td>
<td>19.84%</td>
<td>5.32%</td>
<td>94.68%</td>
<td>09:03:55</td>
<td>32,635,941</td>
</tr>
<tr>
<td>3</td>
<td>pogar</td>
<td>3.22K</td>
<td>153.66M</td>
<td>19.44%</td>
<td>0.36%</td>
<td>99.64%</td>
<td>01:02:34</td>
<td>3,754,743</td>
</tr>
<tr>
<td>4</td>
<td>stereotip</td>
<td>9.27K</td>
<td>80.17M</td>
<td>10.14%</td>
<td>2.05%</td>
<td>97.95%</td>
<td>00:52:35</td>
<td>3,155,360</td>
</tr>
<tr>
<td>5</td>
<td>market</td>
<td>4.23K</td>
<td>51.09M</td>
<td>6.46%</td>
<td>20.71%</td>
<td>79.29%</td>
<td>07:59:40</td>
<td>28,780,901</td>
</tr>
<tr>
<td>6</td>
<td>anton</td>
<td>6.95K</td>
<td>50.61M</td>
<td>6.40%</td>
<td>0.68%</td>
<td>99.32%</td>
<td>00:41:18</td>
<td>2,478,322</td>
</tr>
<tr>
<td>7</td>
<td>urist</td>
<td>864</td>
<td>33.93M</td>
<td>4.29%</td>
<td>1.11%</td>
<td>98.89%</td>
<td>00:08:42</td>
<td>522,727</td>
</tr>
<tr>
<td>8</td>
<td>buhgalter2</td>
<td>3.06K</td>
<td>16.27M</td>
<td>2.06%</td>
<td>4.08%</td>
<td>95.92%</td>
<td>00:56:00</td>
<td>3,360,785</td>
</tr>
<tr>
<td>9</td>
<td>alexv</td>
<td>12</td>
<td>462.50K</td>
<td>0.06%</td>
<td>0.00%</td>
<td>100.00%</td>
<td>09:33:21</td>
<td>34,401,929</td>
</tr>
</tbody>
</table>

**TOTAL**: 49.67K, 790.37M  
**AVERAGE**: 5.51K, 87.81M  
**%TIME**: 3.10%, 96.90%  
**MILISEC**: 41:42:44, 150,164,799  
**ELAPSED TIME**: 04:38:04, 16,684,977  

Generated by sarg-2.2.5 Mar-03-2008 on May/23/2009 06:40
This view shows addresses a particular user is browsing to.
This view shows internal IP addresses.
Shows what Antivirus program the target is running and how often they update.
Shows that the target is running Microsoft Windows and gives hints as to what updates are being installed as well as the frequency of updates.
Analysis of Black Hat Techniques in the Wild
Profiling

- How White Hats get assigned Targets:
  - "Only touch xyz hosts, don't touch abc, those are production"
  - "Hosts 123 we already know are vulnerable, don't worry about those"

- How Black Hats Choose Targets:
  - Source code devs
  - Pen testers
  - Researchers
  - Maintain Control
  - May not yield access immediately
Analysis of Black Hat Techniques in the Wild

• Environment Modeling & Testing
  – White hats test attacks against clients
  – We have seen whole environments mirrored
  – Base mock up on info gathering
    • Match OS, hardware, patch levels, applications
    • Virtualization up to real hardware

• Exploit Development
  – Black Hats write them
  – White Hats use them
Analysis of Black Hat Techniques in the Wild

• **Flexible Environment Testing**
  – Can do vulnerability assessment at leisure
    • Code auditing
      – Double win: 0day + 0wnage
    • Fuzzing
    • Reverse Engineering / Binary Analysis
  – Exploit testing without alerting target
  – One case was 18 months of staging
    • Less than 1 minute of exploitation
    • 5 minutes of data stealing
Analysis of Black Hat Techniques in the Wild

• Examples
  – Attack on Apache.org
  – Attack on Debian.org
  – Attack on Wordpress.com
  – Attack on Comcast.net
  – Attack on Linux Distro
  – Attack on Bank
Analysis of Black Hat Techniques in the Wild

- **Apache.org**
  - Attackers used no exploits. Instead they relied on configuration errors
  - Used a combination of small bugs leveraged against the system to gain
  - Administrative access to the main source repository
  - Patiently waited for root to login.
  - Defaced
Analysis of Black Hat Techniques in the Wild

• Debian.org
  – Attackers used no exploits.
  – SSH Authkey misuse on a system in Japan and a system in the Netherlands
  – Allowed access to the administrative account on debian.org
  – SSHD backdoored and core debian OS source backdoored
  – Was unknown for 6 months
Analysis of Black Hat Techniques in the Wild

- Wordpress.com
  - Attackers used zero day vulnerability
  - Backdoored Live web application
  - Accessed chief source code repository
  - Backdoored source code
  - Was quickly noticed and fixed
Analysis of Black Hat Techniques in the Wild

- **Comcast.net**
  - Attackers used no exploits
  - Attackers Social Engineered Network Solutions into granting them access to Comcast's account
  - Attackers redirected comcast.net domain name to attacker controlled servers
  - Defaced
Analysis of Black Hat Techniques in the Wild

- **Major Linux Distro**
  - Heard of attacker getting in over months
  - Subtlety backdoored distro
    - Introduced bug
    - Matched md5s
    - Able to own any system for 6 months
    - Distro NOT the ultimate target
Analysis of Black Hat Techniques in the Wild

• Hackme Bank
  – Found devel host on separate network
  – Attackers used custom vuln in co-located website
  – Read many files via directory traversal
    • Solaris treats directories like files
      – So you can do cat dir/ and get an ls
  – Discovered copy of every transaction goes over email
  – Copied mail spool via targets own website
  – $$$$$
Analysis of Black Hat Techniques in the Wild

• Air Gap
  – Difficult to hack network w/ smart admins
  – Attackers did recon, read target procedure docs
    • Two networks
      – One online, heavily monitored
      – One offline exact copy cold backup
      – One tape drive machine for copying back and forth
  – Compromised tape system (nothing else vuln)
    • Found 0day in unix TAR
    • Generated a malicious TAR file header
    • Payload wrote malicious binaries into archive
Analysis of Black Hat Techniques in the Wild

- **Air Gap**
  - Exploit had to reload TAR and start untarring from an offset pointing to valid archive
    - Execution continuation
  - Admins eventually moved trojaned backups to “cold” side
  - Attacker made loud (but ineffective) attacks on “hot” side
  - Admins assumed compromise and restored “hot” side from cold backups
    - Thus trojaning their own systems and giving attacker access
Analysis of Black Hat Techniques in the Wild

• Banking backbone
  – Attacker stumbled upon system while doing x25 scans
  – Old ftp / ftp uname & password trick worked for a shell
  – Attacker poked around system and noticed financial transactions
    • LARGE amounts of money
    • Grabbed docs and logged out
  – Turn out to be major banking transaction system
    • All transactions encrypted, but banks would ftp transaction logs to server and store them clear text for balance reconciling
  – By coincidence attacker met system owner in real life
  – Caused no damage, but spent a year hiding
Analysis of Black Hat Techniques in the Wild

• University
  – Attacker compromised system at major university
  – Forensics discovered the compromise
  – Attacker used a kernel rootkit years before common
    • Investigators assumed nation state sponsored attack
    • It wasn’t
    • Rootkit removed
  – Attacker spent 6 – 8 months designing a bios rootkit
  – Re-compromised system and went undetected with new technique
  – Illustrates persistence of some attackers
Black Hat Techniques De-Mystified
Black Hat Techniques De-Mystified

- Few exploits used in attacks
  - Often only 1 exploit needed
  - Rest is captured passwords
  - Trust hijacking
  - Using compromised user's access
    - Datacenter / SSH example
    - authorized_keys infection
Black Hat Techniques De-Mystified

• Few exploits used in attacks
  – Looking like a normal user is hard to detect
    • No shellcode / payloads for IDS to see
    • Traffic looks like normal user activity
  – 0day is priceless
    • Often used when 1day
  – Greater knowledge of system internals is key
  – Attackers know your playbook
    • Blackhats don't do what pen testers do
    • (Unless they want to look like you)
Black Hat Techniques De-Mystified

• Problems attackers run into
  – Secure Data Exfiltration
  – Dangerous Data
    • Mail spools full of viruses
    • Smart targets, documents with attribution call homes
    • Trojaned TAR files
      – Built to overwrite home directories
  – Burn data to CD
  – Read offline on throw away box
    • Avoids above problems
Black Hat Techniques De-Mystified

• Problems attackers run into
  – Retrieving GB's over Tor
  – Download managers not just for warez
  – Scripted Tor wget's
  – POST's instead of GET's
  – Obfuscates logs
  – How to get reverse shells back without attribution?
  – Leaking info during attack (emails / chats)
Black Hat Techniques De-Mystified

• Maintaining Control
  – Data Interception is priority number one.
    • Let the victims do the hacking for you
  – Why use rootkits
    • Detectable
    • Kernel behavior almost always indicates Ownage
  – Better to ensure re-exploitation at will
  – Hide in plain site / look like normal activity
Black Hat Techniques De-Mystified

• Maintaining Control
  – Introduce subtle bugs instead of backdoor binaries
  – Modify source to be vulnerable
    • Harder to detect than blatant backdoor
  – Downgrade applications to vuln versions
  – Re-enable disabled accounts
  – Keep admins & incident response second guessing
    • Flood box with worms & malware if you don't get in
    • Hide in the noise
Black Hat Techniques De-Mystified

• Maintaining Control
  – Example:
  – Machine has VNC installed
  – Replace installed VNC with vulnerable version
    • Authentication bypass
  – Copy registry password so target doesn’t realize software has been updated
  – Persistence with no malware or rootkits to get detected
• **Maintaining Control**
  – Add vulnerable code
  – Example: web apps
    • Take out user input validation
    • Inject your vulnerable code
      – Focus on vague intent
      – Never be *obviously* and *solely* malicious
    • Look for apps with previous vulnerabilities
    • Re-introduce patched bugs
• Maintaining Control
  – More web app examples
  – Add hidden field to HTML form
    • Users detect no change, app performs normally
      \[
      \text{<input type="hidden" name="Lang">}
      \]
  – Edit web app and tie vuln perl code to form field input
    \[
    \text{If defined } $\text{hidden\_field} \{
      \text{open($filename,">$\text{hidden\_field}”);}
    \}
    \]
  – Craft a POST including the hidden field
• Maintaining Control
  – www.target.com/cgi-bin/app.cgi?lang=|cmd|
  – Code will execute your commands
  – Who needs to bind a shell to a port?
  – Unlikely to ever be detected
    • Especially good in big apps
    • Code review can’t ever be sure of maliciousness
    • But some sites replace code every X time-period
  – No rootkits to install
  – Unusual to tripwire all web code
Black Hat Techniques De-Mystified

• Other Attackers
  – Find them on the target
  – Full intrusion analysis
  – Understand what they have done and what they are after
    • Maybe a box you didn’t think was important actually is
  – Model your behavior after them
  – Make your activity look like they did it
  – Find and patch the hole they used to get in
    • Kick them out
Black Hat Techniques De-Mystified

• Other Attackers
  – Example
    • One case found another attacker on same box
    • Had modified login script
    • Exclude logins from attack host from logging
    • Added self as well to same script
Black Hat Techniques De-Mystified

• Protecting Bugs
  – Example
    • Attacker had 0day for commonly used service
    • Rumors circulated
    • Attacker had a colleague leak a different, less reliable but related bug
    • Removed focus from attacker and real bug
      – CMD Exec survived another 4 years
Black Hat Techniques De-Mystified

• Anonymity
  – Hijack wifi
  – Look for default configured u/p WAPs
  – Modify DMZ to get reverse shells back
  – Find web shells on boxes other people hacked
    • Use them as launch pads
    • You didn’t even have to hack them yourself
Black Hat Techniques De-Mystified

• **Anonymity**
  - Tor
  • Hide in the Tor noise
  • Porn, warez & hacking
  • Do all recon possible in Tor or similar
  • Change IP's (Identities) often
  • Use 3rd party web based port scanners
  • Hit target and web tools only from Tor
Black Hat Techniques De-Mystified

• **Anonymity**
  – Tor C&C
    • See Metaphish Talk
    • 100% True SSL encrypted
    • Cross platform
      – Mono
      – Linux & Windows with same binary
  • Communicates using Tor hidden services
  • Even if target:
    – Reverses backdoor
    – Has 100% packet capture
    – They cannot trace it back to source
Black Hat Techniques De-Mystified

• **Anonymity**
  – Covert communications
  – Attackers use strange covert communications
  – Example
    • Edonkey p2p with crypto enabled appears to simply be SSL traffic
    • Some attackers known to use this for file transfer and communications
    • In one case TCP over edonkey
  – Have seen attackers using twitter, gmail and msn messenger for command and control of compromised systems
Never Caught
Never Caught

- Anti-forensics & Law Enforcement
  - Cell phone alibi
    - Place phone in desired location away from attack
    - Have call made to phone
    - Have phone answered
      - Accomplices bring complications
    - Auto answer programs for smart phones
    - When phone records are pulled:
      - Location + call record "prove" your location
  - Buy a movie ticket & leave movie early
  - Whole field of study: Alibiware
Never Caught

- Anti-forensics & Law Enforcement
  - Reset every timestamp on system to same date
    - Timestomp
  - Encase exploits

- Memory only & staged C&C
  - Just enough code to receive next chunk from network
  - True SSL
  - Need full packet capture + break SSL to get C&C analysis
  - No real malware on disk to RE
Never Caught

• Data protection & destruction
  – Attackers have to protect their data from other attackers and law enforcement
    • Some attackers encrypt all data with complex key
    • One group of attackers built a drive “chipper”
    • 1 ½ horse power motor from a metal router
    • Metal router blades
    • Result a giant bin full of no bigger than ½ inch square drive parts
    • Good luck getting forensic data
What does all this mean?
What does all this mean?

- Attackers are determined
  - They will not stop
- Attackers are extremely patient
- Only have to succeed once
- Understand how an attacker thinks
- Know your Enemy
- Test everything
  - Small bugs yield Big bugs

- Black Hats are not all powerful
  - They just know more tricks
- Many pen testers are providing unrealistic tests
- Full scope best value
What does all this mean?

• What can you do?
  – Proper Training
  – Investigate Reports
  – Identify Targets
  – Predict Attackers
  – Proactive Defense is best
  – Defense is not System Administration
  – Properly Mitigate Risk
  – Learn from other peoples mistakes
  – Open Discussion