



2nd Generation Honeyclients

Robert Danford
SANS Internet Storm Center

Research funded in part with a grant from StillSecure (<http://www.stillsecure.com>)

Author can be reached at:

rdanford@isc.sans.org

<http://handlers.dshield.org/rdanford/>

Download this presentation at:

http://handlers.dshield.org/rdanford/pub/2nd_generation_honeyclients.ppt

Download related paper at:

http://handlers.dshield.org/rdanford/pub/2nd_generation_honeyclients.pdf

All slides copyright 2006 Robert Danford

What are Honeyclients?

High-interaction (active)
client-side
honeypots

used for detecting and characterizing
malicious sites by driving a system
in a way that mimics human users



2nd Generation Honeyclients

2

May be physical or virtual

Virtual honeypots may have reduced effectiveness due to anti-vm techniques

Most honeyclients automate browsers and interact with websites.

However honeyclients which interact with P2P networks, instant messaging, etc are inevitable if not in current development.

Low/Medium/High Interaction

Low	Transport layer virtualization
Medium	Application layer virtualization
High	Real, vulnerable systems



2nd Generation Honeyclients

3

Examples by Classification

Low Interaction Honeybots

- Honeyd: <http://www.citi.umich.edu/u/provos/papers/honeyd.pdf>
- Deception Toolkit (DTK): <http://all.net/dtk/index.html>
- LaBrea Tarbit (a sticky honeypot): <http://labrea.sourceforge.net>

Medium Interaction Honeybots: www.pixel-house.net/midinthp.pdf

- Mwcollect: <http://www.mwcollect.org>
- Nepenthes: <http://nepenthes.mwcollect.org>
- Multipot: <http://labs.iddefense.com/labs-software.php?show=9>
- Full system using jail or chroot

High Interaction Honeybots

- Actual operating system (may be running in a virtual machine)
- Netbait: <http://www2.netbaitinc.com:5080/>
- Honeynets: <http://www.honeynet.org/papers/honeynet/>
- Mantrap/Symantec Decoy Server
- Collapsar
- Honeyclients

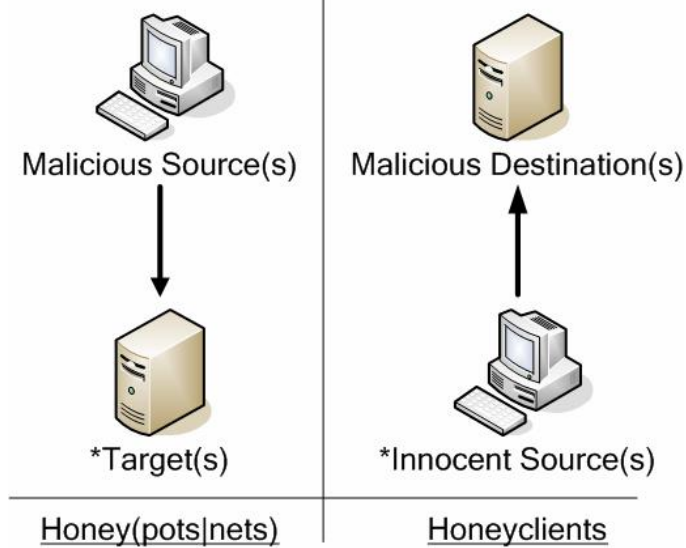
Trade-offs

- Speed
- Ease of
 - Implementation
 - Maintenance
 - Reuse
 - Detection (fingerprinting)
- Depth of information gathered
- Reality vs. simulation
- Resources required



This space intentionally left blank.

Server-side vs. Client-side



* - Denotes location under control for logging and analysis

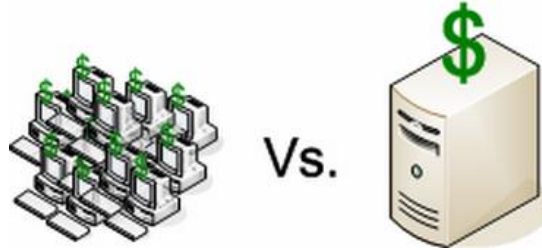
Traditional honeypots are passive and server-side.

They provide a wide variety of benefits:

- Worm detection
- Insider-abuse detection (also honeytokens)
- Malware capture (medium-interaction honeypots such as nepenthes are particularly good at this)
- Security research
- New exploit detection

Why client-side is so important

- Threats triggered by end-user behavior
- Security is fundamentally a human problem
- Criminal focus on soft-targets



2nd Generation Honeyclients

6

This space intentionally left blank

Honeyclient Uses

- Evaluating/characterizing web sites
- Testing endpoint security
- Detecting zero-day browser exploits
- Mapping malicious neighborhoods
- Obtaining unique malware and exploit samples



2nd Generation Honeyclients

7

See MS Strider HoneyMonkey project for results on zero-day detection

Honeyclients allow for the capture of malware and exploit samples that are not easily obtained by other methods.

Honeyclients developed out of honeypot research. They are basically active reverse-honeypots:

Usually deployed as farms to increase the number of websites that can be reviewed or to increase the amount of review that can be done on a single site in a reasonable amount of time

Well-suited for researching a number of current threats directed at clients rather than servers such as:

- Browser exploits
- Drive-by downloads
- Adware/spyware
- Trojan downloaders
- Phishing

When real systems are used configuration, security policies, and patch levels can be validated against actual malicious websites.

Examples

- Drive-by downloads
- Adware/spyware
- Exploitation websites
- Phishing?
- Typo-squatting
- Zero-day exploits against browsers



For further reading:

http://en.wikipedia.org/wiki/Drive-by_download

<http://www.benedelman.org/spyware/ftc-031904.pdf>

http://www.mnin.org/write/2005_trimode.html

<http://www.websensesecuritylabs.com/alerts/alert.php?AlertID=451>

<http://research.microsoft.com/URLTracer/>

<http://www.securityfocus.com/news/11273>

Honeyclients vs. Crawlers

Honeyclients

- Vulnerable to attack
- Utilize a mechanized browser when surfing
- Must be monitored to detect compromise
 - Blackbox (MS Strider)
 - Integrity checks
 - Scans (AV, AS, etc)
 - Intrusion Detection
 - Sandbox (Sandboxie)

Crawlers

- Not supposed to be compromised
- Crawlers programmatically surf websites to retrieve content
- Simulation can be used to determine if content is malicious (sandbox)



2nd Generation Honeyclients

9

Some crawlers could be classified as low-interaction (passive) client-side honeypots if they are capable of characterizing the retrieved content as malicious or contain a sandbox-like environment to simulate compromise/infection.

Dan Hubbard (Websense) also describes Hybrid honeyclients which are a mix between active and passive.

The active content environment is simulated (JavaScript, VB, etc) not just the browser. This is perhaps similar to the application layer virtualization concept in Medium-interaction honeypots such as mwcollect and nepenthes.

Sandboxie is a great new tool from Ronen Tzur (tzuk)
And can be downloaded from: <http://www.sandboxie.com>
(Thanks William!)

Issues with Crawlers

- Simulation
 - Exploit may not trigger
 - Active/dynamic content
 - Chain reactions
 - Secondary vulnerabilities
- Ease of detection
 - Fingerprinting
- Maliciousness detection
 - Signatures
 - Interpretation



This space intentionally left blank.

Issues with Honeyclients

- Speed
 - More complexity = slower
- Stability
 - Infected systems are slow
- Maintenance
 - Reset after infection
- Maliciousness detection
 - Sandbox, IDS, scanners



This space intentionally left blank.

Projects Utilizing Honeyclient or Crawler Technology

- MS Strider HoneyMonkey (Microsoft Research)
- Honeyclient.org (Kathy Wang)
- Mitre Honeyclient Project (Mitre)
- Client-side Honeybots (Univ. of Mannheim)
- Collapsar/Reverse Honeyfarm (Purdue Univ.)
- Phileas (Webroot)
- Websense (Hubbard)
- SiteAdvisor (McAfee)



2nd Generation Honeyclients

12

URLs for projects referenced above:

1. <http://research.microsoft.com/HoneyMonkey>
2. <http://honeyclient.org>
3. <http://honeyclient.mitre.org>
4. <http://pi1.informatik.uni-mannheim.de/diplomas/show/27>
5. <http://www.cs.purdue.edu/homes/jiangx/collapsar/>
6. <http://www.webroot.com/resources/phileas/detail.html>
7. http://www.websensesecuritylabs.com/resource/PDF/toorcon_sep2005.pdf
8. <http://www.eweek.com/article2/0,1895,1933818,00.asp>

Projects Utilizing Honeyclient or Crawler Technology Cont'd

- StillSecure/Pezzonavante (Danford)
- SPECTRE (Sunbelt)
- Shadow Honey pots (Anagnostakis)
- Email quarantine systems (Columbia Univ.)
- Spycrawler (Univ. of Washington)
- XPLIntel (Exploit Prevention Labs)
- Irish Honey net Project (Espion)



2nd Generation Honeyclients

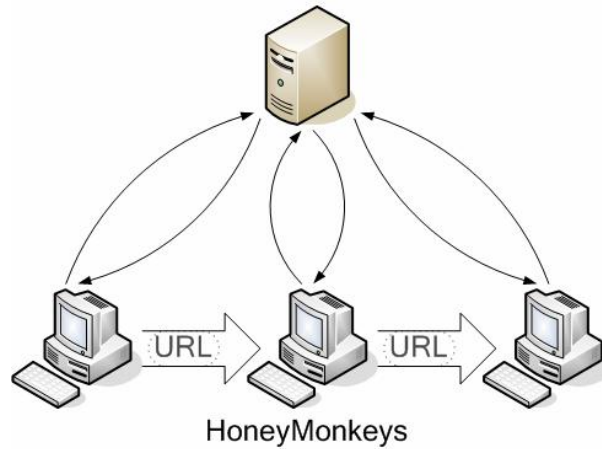
13

URLs for projects mentioned above:

1. http://www.infoworld.com/article/06/04/14/77378_16OPsecadvise_1.html
2. <http://www.eweek.com/article2/0,1895,1788878,00.asp>
3. <http://www.usenix.org/events/sec05/tech/anagnostakis.html> (Their work mechanized firefox instead of MS Internet Explorer)
4. <http://www.cs.columbia.edu/~angelos/Papers/2005/email-worm.pdf>
5. <http://www.cs.washington.edu/homes/gribble/papers/spycrawler.pdf>
6. <http://www.infosecwriters.com/hhworld/collage/ssxpl.htm>
7. <http://www.net-security.org/secworld.php?id=3715>

MS Strider HoneyMonkey Project

malicious website



HoneyMonkeys

Less patched \longrightarrow More patched



2nd Generation Honeyclients

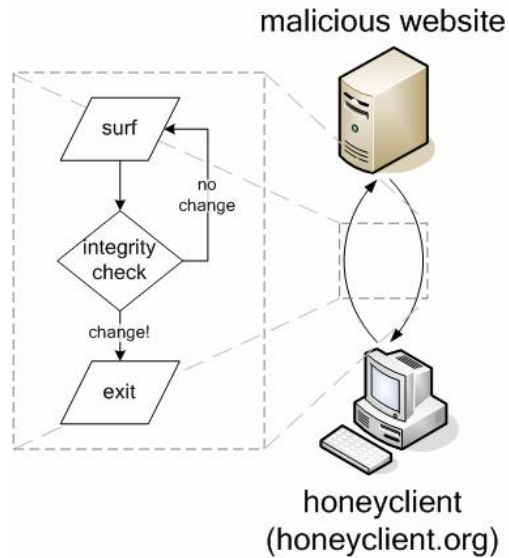
14

GOALS:

- Detect zero-day attacks against fully patched Microsoft systems
- Map out and research malicious neighborhoods and pursue criminals using legal and law enforcement means

<http://research.microsoft.com/HoneyMonkey/>

Honeyclient.org Honeyclient



2nd Generation Honeyclients

15

GOALS:

- Proof of concept open-source honeyclient
- Encourage community participation and improvements (ex. Email Honeyclient contributed by Dublin City Univ.)
- Directly associate a website visit with specific changes to the endpoint

<http://honeyclient.org>

Issues to Overcome

- Constant supply of URLs
- Preventing infected clients from infecting the planet (honeywall)
- Surf tracking (URL Server)
 - Results from each visit
 - Coordinate across clients
 - Limited retries
- Correlating infections
- Avoid being blacklisted



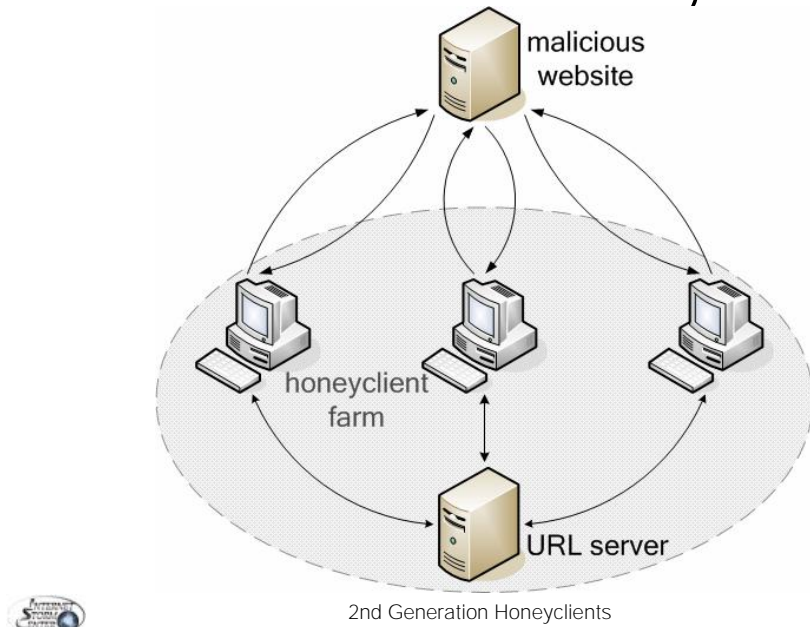
2nd Generation Honeyclients

16

Pezzonavante utilized coordinated surfing across a farm of honeyclients for several reasons:

1. Insure all clients visited the same sites. Web site availability is not consistent due to takedowns, etc
2. Take full advantage of squid caching
3. Expose clients with different security stances to the same potentially malicious content

Pezzonavante Honeyclient



GOALS:

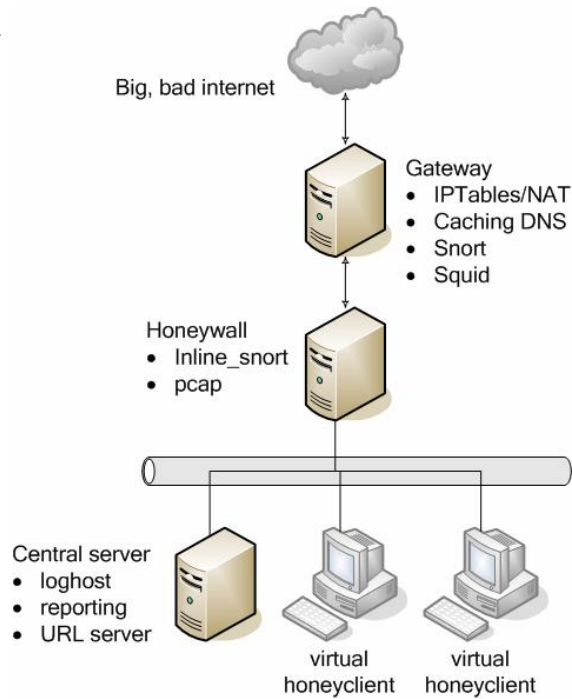
- Create a honeyclient capable of coordinated surfing at high speed.
- Validate the security policies of endpoints in the face of live infections
- Capture novel malware samples to share with the security community
- Identify malicious URLs hosting exploits or malware and coordinate with groups capable of having them shut down.

http://www.infoworld.com/article/06/04/14/77378_16OPsecadvise_1.html

Deployment Experience

October 2005 - March 2006

- 200,000 URLs surfed
- 7 million links harvested
- 600+ virus infections
- 750+ spyware-related events
- 1,500 malware samples
- 500+ malicious URLs submitted for takedown



18

<http://www.stillsecure.com/endpointindex> (site no longer active)

Issues Found

- Speed
- Coordination
- Correlation
- Information Overload
- Candidate URLs
- Anti-VMware techniques

Infected PCs are slow and unstable. Duh!



2nd Generation Honeyclients

19

Further reading:

<http://www.honeynet.org/tools/sebek/>

<http://ntsyslog.sourceforge.net/>

<http://safety.live.com/site/en-us/article/slowpc.htm>

Synchronous vs. Asynchronous honeyclients

Pezzonavante was able to achieve much higher surf rates compared to the honeyclient development project (honeyclient.org) in part by not integrity checking after every URL surfed.

Infections were primarily detected and reported on at month end. Many more malware samples and malicious URLs were found at the expense of failing to alert on some malicious sites.

Synchronous honeyclients can be sped up through brute force (aka using clusters of high-speed computers)

Characterizing URLs

- Potentially malicious websites need to be identified in advance (guided search)
- Avoid surfing .mil, .gov, and froogle links all day
- 10x more URLs were harvested from links than could be surfed each month.
- Needle in a haystack problem. Must prioritize.

Month	URLs Surfed	Candidate URLs
1	235,000	2,000,000
2	470,000	3,765,000
3	705,000	5,530,000



2nd Generation Honeyclients

20

If there are ten billion web pages and 0.02% of them are malicious, we're trying to stumble across any of those two million pages.

Further reading:

http://news.netcraft.com/archives/2006/05/09/may_2006_web_server_survey.html

<http://www.sims.berkeley.edu:8000/research/projects/how-much-info/internet.html>

<http://www.internetworldstats.com/stats7.htm>

<http://blog.searchenginewatch.com/blog/041111-084221>

Methods for Determining Candidate URLs

1. Compare IP/hostname against blacklists
2. Filename ends in an executable suffix (*.scr, *.exe, *.pif)
3. Known-bad strings (ie0502.htm, cartao, cmd.txt)
4. Obfuscated URLs
5. Known redirectors (from previous squid logs)
6. McAfee SiteAdvisor ranking
7. Site logged in the Norman Sandbox
8. Site or URL substring shows up in virus descriptions



2nd Generation Honeyclients

21

Reference URLs:

1. <http://dmoz.org/Computers/Internet/Abuse/Spam/Blacklists/>
2. <http://antivirus.about.com/od/securitytips/a/fileextview.htm>
3. .Either from experience or sources like snort rules and mod_security rules (see below)
4. <http://www.rain.org/~mkummel/stumpers/08dec00a.html>
5. <http://giannis.stoilis.gr/software/mysar/> (looks pretty cool. Watch for HTTP 3xx codes)
6. <http://www.siteadvisor.com/analysis/>
7. http://sandbox.norman.no/live_5.html
8. <http://www.f-secure.com/v-descs/randon.shtml> (example)

Further reading:

<http://www.microsoft.com/technet/security/tools/urlscan.msp>

<http://www.modsecurity.org/>

<http://www.modsecurity.org/download/modsecurity-rules-current.tar.gz>

<http://www.bleedingsnort.com/cgi-bin/viewcvs.cgi/signs/WEB/>

URL Sources

- URLs harvested from unsolicited email
- Google API
- Harvested links
- SANS ISC URL list
- Blacklists



2nd Generation Honeyclients

22

URLs referenced above:

1. http://www.postfix.org/SMTPD_POLICY_README.html
2. <http://www.amazon.com/gp/product/0596004478/102-5968845-8340159?v=glance&n=283155>
3. Link harvesting performed by the honeyclient software
4. <http://isc.sans.org/urllist.php>
5. <http://www.spywarewarrior.com/uiuc/resource.htm>

Detecting Malicious Activity

Pezzonavante used a hybrid,
asynchronous approach to detection

- Osiris integrity checking
- Security tool scans
- Snort network IDS alerts
- Traffic analysis
- Snapshot comparisons



2nd Generation Honeyclients

23

This approach was in contrast to blackbox and sandbox techniques used in other projects (MS Strider HoneyMonkey)

Blackbox notes:

The MS Strider project used a number of tools they developed to determine infection:

- Strider Flight Data Recorder - Records every file and registry read or write
- Strider Gatekeeper - Detects any unauthorized hooking of Auto-Start Extensibility Points (ASEPs)
- Strider GhostBuster - Detects stealth malware that hide processes and ASEP hooks

Other tools:

- SysInternals Filemon - Log all file operations
- SysInternals Regmon - Log all registry access
- Spybot S&D teatimer - Intercept and log all registry operations
- Sandboxie - Full program sandbox
NEW <http://sandboxie.com/>

Sandboxes and Integrity Checking

- CWSandbox -
- Sandboxie -
<http://www.sandboxie.com/>



2nd Generation Honeyclients

24

Sandboxes:

Several projects also use a sandbox approach where they pre-define every acceptable system activity and trigger on outside activities such as a process writing files outside the sandbox, an unauthorized process executing, unauthorized network communication, etc.

Integrity Checking:

A number of tools are available for integrity checking such as:

Regshot - registry snapshots

Osiris - general integrity checking (filesystem, users, etc)

Manual registry checks (PERL - honeyclient.org)

MD5 checksums of critical system files (PERL - honeyclient.org)

Scans:

A number of security tools are useful in scanning a system for malicious changes and infections:

Norton Anti-Virus

HijackThis

Trend Micro Anti-Spyware

Spybot Search & Destroy

F-Secure Blacklight Beta

SysInternals RootkitRevealer

Lavasoft AdAware

Microsoft Anti-Spyware/Windows Defender

Norman Virus Control

MyNetWatchman SecCheck

Vulnerability Scanners (ex. StillSecure VAM, Nessus, etc)

Intrusion Detection and other observations:

Systems can be watched for signs of intrusion using traditional methods:

Note: log review is made easier by using specific backwater networks to host honeyclient farms.

Snort Network IDS

Win32 Eventlog monitoring

DNS query monitoring

Network traffic analysis

Squid log analysis

Blackbox and sandbox techniques appear to be the most reliable.

Thorough integrity checking methods couple with intrusion detection methods and local scanning would be next best.

Integrity Checking

Osiris

compare time: Tue Oct 25 22:57:40 2005
host: victim5
scan config: master (9ad9b7e7)
log file: 173
base database: 19
compare database: 20

```
[203][victim5][new][c:\windows\dlgb.exe]
[203][victim5][new][c:\windows\extract.exe]
[203][victim5][new][c:\windows\id.exe]
[203][victim5][new][c:\windows\ieupdate.dat]
[203][victim5][new][c:\windows\system32\appwiz.dll]
[203][victim5][new][c:\windows\ts.exe]
[203][victim5][new][c:\windows\v010101.exe]
[203][victim5][new][c:\windows\wupdt.exe]
```



2nd Generation Honeyclients

25

<http://www.hostintegrity.com/osiris/>

Osiris was extremely useful. Policies and logs were managed from a central server. Emailed reports helped with endpoint monitoring.

Anti-Virus

Running an anti-virus product after the fact will produce some results

Date	Filename	Virus Name	Status
11/17/2005 17:04	yes[1].htm	Trojan Horse	Infected
11/17/2005 17:04	uol[2].exe	Download.Trojan	Infected
11/17/2005 17:04	n3[1].exe	W32.Kelvir	Infected
11/17/2005 16:59	ass[1].chm	Downloader.Trojan	Infected

However, many are missed...



2nd Generation Honeyclients

26

At the end of each month of my project we would run security tools one after the other.

Each tool was updated beforehand and then allowed to detect and clean/quarantine/etc whatever it found.

After every monthly run we were still detecting new infections after running the fourth or fifth tool.

A little scary in my opinion.

Intrusion Detection (Snort)

NIDS was most helpful in monitoring for post-infection behavior.

However, occasional gems were found.....

```
[**] [1:2436:5] <eth5> WEB-CLIENT Microsoft wmf metafile access [**]
```

```
192.168.1.205:51372 -> 85.255.115.196:80 TCP TTL:64 TOS:0x0  
ID:49261 IpLen:20 DgmLen:609 DF
```

```
***AP*** Seq: 0xBC5B3A5B Ack: 0xDB156A2C Win: 0x5B4 TcpLen: 32  
[Xref => http://www.securityfocus.com/bid/10120]
```



This space intentionally left blank

Squid is Your Friend

Log entry for site access referenced in previous slide

```
113402032.379 376 192.168.1.205 TCP_MISS/200 1315
GET http://196.regvista.com/ie0601e.wmf -
DIRECT/85.255.115.196 video/unknown
[HTTP/1.1 200 OK
Date: Thu, 02 Feb 2006 11:21:12 GMT
Server: Apache/2.0.53 (Fedora)
Accept-Ranges: bytes
Content-Length: 1024
Connection: close
Content-Type: video/unknown]
```



2nd Generation Honeyclients

28

This space intentionally left blank

Network Traffic Analysis (IPTables)

- Basic visualization needs similar to other honeynet projects
- New visualization tools needed to observe near real-time activity on the client

```
Oct 20 19:36:41 localhost kernel: OUTBOUND TCP: IN=br0 OUT=br0  
SRC=192.168.1.205 DST=1.1.1.1 LEN=48 TOS=0x00 PREC=0x00 TTL=128  
PROTO=TCP SPT=1229 DPT=6667 WINDOW=64240 RES=0x00 SYN
```



Excellent honeypot data visualization

<http://www.philippinehoneynet.org/data.php>

Further reading:

<http://www.securityfocus.com/infocus/1854>

Anti-honeyclient Methods

1. Blacklisting
 - Try to "look" normal and not get blacklisted
 - Distributed honeyclient farms
2. Dialog boxes
 - GUI automation needed (ex. Windpysend)
3. Anti-crawler techniques
4. Time-bombs
 - Wait 10 sec in case of delayed exploit
5. Page-close events
 - Load a blank page to trigger event (delayed exploit)



2nd Generation Honeyclients

30

1. Many of the issues listed will get you blacklisted
2. <http://users.swing.be/wintclsend/windpysend/>
3. http://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Shah.pdf

Further reading:

- <http://www-i4.informatik.rwth-aachen.de/lufg/research/projects/honeynet/material/2004-NoSEBrEaK.pdf>
- <http://securityfocus.com/infocus/1826>
- <http://securityfocus.com/infocus/1828>
- <http://www.securityfocus.com/infocus/1803>
- <http://www.securityfocus.com/infocus/1805>
- <http://www.tracking-hackers.com/junkyard/paper/Holz-2005-DHO.pdf>
- <http://www.leekillough.com/robots.html> (spider traps)

Anti-honeyclient Methods Cont'd

6. Non-deterministic URL behavior
 - Pool stats with other farms. Overlap surfing
7. Links no human would click
 - Background color hyperlinks
 - IMG links with "don't click" on them
8. Timing analysis
9. Surf behavior
 - Timing analysis
 - Paths through a site
 - Depth-first vs. breadth-first
 - Referer information (deep linking)



Some of these issues apply more to crawlers.

When we mechanize Internet Explorer web sites are surfed fairly normally.

The main issue is how harvested links are queued for surfing.

When the links are added to the candidate pool for possible surfing at a later date instead of being surfed immediately the activity is not spider-like.

Anti-honeyclient Methods Cont'd

10. Dynamic and relative URLs
 - JavaScript \$* & #*
11. Cookies
12. Session IDs
13. Encoded URLs, foreign character sets
14. URL redirection



2nd Generation Honeyclients

32

10. Presents an issue for link harvesting. Full URLs need to be fixed up for storage in the database.
11. Cookie support is not a problem for a mechanized browser, but if harvested links are surfed at a later date or by a different honeyclient farm then stateful information such as cookies may need to be retained and distributed with the URL list.
12. Session IDs present a similar problem
13. UTF-8 support is needed (<http://en.wikipedia.org/wiki/UTF-8>)
IDN may be an issue in the future (<http://www.idnnow.com/index.jsp>)
URL encoding (<http://427.cgisecurity.com/lib/URLEmbeddedAttacks.html>)
14. There are several methods of redirecting browsers. Mechanized IE should follow along. Careful squid logging and analysis is required (http://en.wikipedia.org/wiki/Url_redirection)

Further reading:

<http://us2.php.net/session>

Malware Analysis Evasion

- Current trend in certain malware code-bases for detecting debugger or virtual machine environments
- More study required to determine what percentage of infections virtual honeyclients may miss
- Physical machines plus a disk imager like Ghost may be needed



2nd Generation Honeyclients

33

Check out Paul Barford's paper on botnet architectures for more information on malware utilizing vmware and debugger detection techniques. (agobot, etc). If our goal is to pick up the newest/nastiest malware on the internet with honeyclient farms we can't afford to have malware aborting execution and hindering analysis. The upside is that as virtualization becomes more widespread, malware authors will have to decide on the trade-off between a lower install base (no virtual machines) or risking making malware researchers lives easier.

http://www.cs.wisc.edu/~pb/botnets_final.pdf

http://searchopensource.techtarget.com/columnItem/0,294698,sid39_gci1176065,00.html

<http://www.virtualization.info/2005/11/microsoft-windows-hypervisor-and-amd.html>

Anti-VMware and VMware Detection Methods

1. Nopill (Smith)
2. Vmdetect (Lallous)
3. Redpill (Rutkowska)
4. Scoopy Doo (Klein)
5. Jerry (Klein)
6. Vmtools (Kato)



2nd Generation Honeyclients

34

1. <http://www.offensivecomputing.net/files/active/0/nopill.cpp>
2. <http://www.codeproject.com/system/vmdetect.asp>
3. <http://invisiblethings.org/papers/redpill.html>
4. http://www.trapkit.de/research/vmm/scoopydoo/scoopy_doo.htm
5. <http://www.trapkit.de/research/vmm/jerry/jerry.htm>
6. <http://chitchat.at.infoseek.co.jp/vmware/vmtools.html>

Further reading:

Thorsten Holz and Frederic Raynal

<http://www.tracking-hackers.com/junkyard/paper/Holz-2005-DHO.pdf>

<http://www.cs.nps.navy.mil/people/faculty/irvine/publications/2000/VMM-usenix00-0611.pdf>

<http://www.phrack.org/fakes/p63/p63-0x09.txt>

<http://honeynet.rstack.org/tools/vmpatch.c>

<http://www.securityfocus.com/archive/119/349385>

<http://www.offensivecomputing.net/files/active/0/vm.pdf>

Malware Analysis Frameworks

- Analysis requires automation
- Sandboxes and fully instrumented lab networks
- Tools for building your own



<http://handlers.dshield.org/rdanford/maf/>

The Future

- Data aggregation
- Data sharing
- Distributed Honeyclient Farms
- Correlate honeyclient and honeynet data
- Analysis (SANS ISC, CastleCops PIRT)
- Coordinated take-downs



<http://isc.sans.org>

<http://wiki.castlecops.com/PIRT>

Resources



2nd Generation Honeyclients

37

Honeywall <http://www.honeynet.org/tools/cdrom/>

Honeynet Alliance <http://www.honeynet.org/alliance/>

Honeyd <http://www.honeyd.org>

http://www.philippinehoneynet.org/docs/Honeypot101_history.pdf

Mwcollect Alliance <http://www.mwcollect.org/>

Norman Sandbox <http://sandbox.norman.no/>

Sandnets: <http://www.lurhq.com/truman/>

Castlecops PIRT <http://wiki.castlecops.com/PIRT>

Win32::IE::Mechanize: <http://search.cpan.org/dist/Win32-IE-Mechanize/>

Postfix Policy Servers: <http://sourceforge.net/projects/p-ppolicyserver/>

Google APIs: <http://code.google.com/apis.html>

Google Hacks:

<http://www.amazon.com/gp/product/0596008570/102-5968845-8340159?n=283155>

Spidering Hacks:

<http://www.amazon.com/gp/product/0596005776/102-5968845-8340159?n=283155>

<http://radlab.cs.berkeley.edu/wiki/2006WinterRetreat/AdaptiveReplay>

<http://radlab.cs.berkeley.edu/w/uploads/4/47/Roleplayer-radlab-retreat-w06.ppt>

<http://research.microsoft.com/URLTracer/>

<http://www.sandboxie.com/>