www.ists.dartmouth.edu/classroom/crs/ndecoys/ndecoys.v0.9.ppt

Network Decoys

Perhaps if we confuse them enough, they'll just go away... ;-)

Goals

- Hide and lie about the structure and composition of our networks in order to confuse and delay attackers
- Do so in a way that legitimate users of our systems and servers are not hindered.

Types of Decoys

- Tarpits
- Router-responders
- Kernel modifications and settings
- Fake services
- Full honeypots

Decoy Tools

- LaBrea
- IPTables
- IPPersonality
- WinX registry settings
- Linux kernel parameters
- Portsentry
- VMWare
- User-Mode Linux

LaBrea

- “Sticky” Honeypot
- Holds machines for days or weeks.
- Effective at holding dumb scanning worms
- http://www.hackbusters.net/LaBrea.html

LaBrea install

- rpm –Uvh ftp://ftp.stearns.org/pub/wstearns/la
- Read documentation in /usr/share/doc/labrea-2.3/LaBrea.README
- Edit /etc/labrea.conf, uncomment: #LABREAZ="-z"
- Add any other needed options.

LaBrea warnings

- Run only on the network segment that holds the IPs you're tarpitting
- Use the exclude files in /etc to list any IP's in use so LaBrea will never fight machines that aren't always there.

IPTables

- Is that system there or not?
- Filtering out icmp echo requests and replies to existing machines.
- Responding for nonexistant machines
- Sending Resets in response to malicious packets (use with care!)

IPTables – filtering icmp echos

- iptables –I INPUT –p icmp –icmp-type echo-request –j DROP
- iptables –I INPUT –p icmp –icmp-type echo-reply –j DROP

IPTables – limiting outbound traffic

- iptables –I OUTPUT –p icmp –icmp-type echo-reply –j DROP
- iptables –I OUTPUT –p icmp –icmp-type time-exceeded –j DROP
- iptables –I OUTPUT –p icmp –icmp-type fragmentation-needed –j ACCEPT
- iptables –I OUTPUT –p icmp –icmp-type destination-unreachable –j DROP

IPTables – responding for nonexistant machines

```
for OneHost in 1.2.3.4 1.2.3.5 ; do
    iptables –I –p udp –d $OneHost –j REJECT \
    --reject-with port-unreach
    iptables –I –p tcp –d $OneHost –j REJECT \
    --reject-with tcp-reset
    iptables –I –p icmp –d $OneHost –j DROP
    iptables –I –d $OneHost –j REJECT \
    --reject-with proto-unreach
done
```

IPTables – shutting down malicious connections

```
iptables –A FORWARD –p tcp –d 1.2.3.0/24 \
--dport 23 –tcp-flags ACK ACK –m string \
--string "r00t" –j LOG
iptables –A FORWARD –p tcp –d 1.2.3.0/24 \
--dport 23 –tcp-flags ACK ACK –m string \
--string "r00t" –j REJECT
```

OS Fingerprinting

- Queso
- Nmap
- P0f (passive OS fingerprinting)
- Ettercap

OS Fingerprinting - One side effect

- If you change these fields to ones not used by an existing OS, you've uniquely fingerprinted your own box. ☺

IPPersonality

- Modifies characteristics to confuse nmap
- Additional modules to iptables
- http://ippersonality.sourceforge.net
- Can emulate a different OS quite readily if run on actual system
- Can at least confuse nmap if run on an intermediate router.

WinX registry settings

- Confusing p0f
- All of the following keys can be modified with regedit.
- All keys except MSS can be found under:
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
- We can’t modify Noop or Packet size

Window Size

- “GlobalMaxTcpWindowSize”=dword:00010000
- “TcpWindowSize”=dword:00010000

Time to live

- “DefaultTTL”=dword:00000030

Maximum segment size

- Under: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\<Adapter>\
- “MTU”=dword:000005dc
- “EnablePMTUBHDetect”=dword:00000000
- “EnablePMTUDiscovery”=dword:00000000

Don’t Fragment

- Make the same PMTU changes as in the last slide to disable the DF flag.

Window Scaling

- “Tcp1323Opts”=dword:00000001

Sack OK

- “SackOpts”=dword:00000000

Linux kernel parameters

- Configured by writing values to /proc/sys/net/ipv4
- Additional documentation in /usr/src/linux/Documentation/networking/ip-sysctl.txt
- Again, we can’t modify noop or packet size

Window size

- tcp_window_scaling (0 or 1)
- tcp_app_win

Time to Live

- ip_default_ttl (default 64)
- To set a new default TTL:
- echo 32 >/proc/sys/net/ipv4/ip_default_ttl

Maximum segment size

- We can indirectly influence this by modifying the MTU (default: 1500 for ethernet).
- To set MSS to 1000, use MSS=MTU-40
- ifconfig eth0 mtu 1040

Don’t Fragment

- ip_no_pmtu_disc (default 0 = false; this means Path MTU discovery is turned ON)
- To stop linux from performing Path MTU discovery, turning OFF the DF flag:
- echo 1 >/proc/sys/net/ipv4/ip_no_pmtu_disc

Window Scaling

- tcp_window_scaling (default 1 = true)
- To disable window scaling:
- echo 0 >/proc/sys/net/ipv4/tcp_window_scaling

Sack OK

- tcp_sack (default 1)
- To turn off sacks:
- echo 0 >/proc/sys/net/ipv4/tcp_sack

Portsentry

- Listens on a large number of ports.
- Looks for portscans, or connections to vulnerable ports
- Can automatically block the scanner via route, ipfwadm/ipchains, or tcp wrappers
- http://www.psionic.com/abacus/port

Honeypots

- Full functioning systems that tie up the attackers attention, allowing you to monitor their actions and capture their tools. ☺
- Not a defense tool, but a passive monitoring tool.
- http://www.honeynet.org

Honyepots – Physical systems

- Physical box with full OS
- Advantages: exactly identical to an operational system
- Disadvantage: expensive, time consuming to set up and perform analysis

Honeypots – VMWare

- VMWare simulates an X86 physical box in software, allowing you to run a virtual machine and X86 OS underneath Windows or Linux
- Advantage: reduced hardware cost
- Disadvantages: inability to make custom changes to the environment, cost of Vmware, cpu overhead, X86 only.
- http://www.vmware.com

Honeypots – User-Mode Linux

- Version of the linux kernel that runs inside an existing linux installation
- Advantages: low cpu and memory overhead, easy to set up and analyze, reduced hardware cost
- Disadvantages: Linux only on host and client
- http://user-mode-linux.sourceforge.net

Credits and Thanks

- Chris Brenton
- Matt Scarborough
- Authors of the respective packages