

St. Louis Security Group
July 25th, 2007

INTRUSION DETECTION WITH SNORT

Introductions

- ◎ Chris Byrd, CISSP
 - Senior Security Engineer
 - chris@riosec.com
 - Presentation will be posted at www.riosec.com

Why IDS

- ⦿ Intrusion Detection Systems a Market Failure – Gartner, Inc. (WRONG!)
- ⦿ IDS is like having an alarm system for your physical warehouse
 - Can't prevent by itself, but allows a response
- ⦿ IDS is a useful tool for experienced operators
- ⦿ Adds no value without added human intelligence
- ⦿ IDS is useful for 24x7 Security Operation Center (SOC) real-time analysis or post-mortem analysis

Flavors of IDS

- ⦿ Host based Intrusion Detection (HIDS)
 - OSSEC – www.ossec.net
 - Osiris - osiris.shmoo.com
- ⦿ Network based Intrusion Detection (NIDS)
 - Snort – www.snort.org
 - Bro – www.bro-ids.org
 - Prelude – www.prelude-ids.org

Network Security Monitoring

- ◎ Championed by Richard Bejtlich
 - taosecurity.blogspot.com
- ◎ Idea: Collect more data
 - Alerts – information about attack
 - Flow data – information about all sessions, allows correlation
 - Full content – best way to know for sure if an attack was succesful
- ◎ Idea: Classify every alert
 - Allows for reporting and metrics
- ◎ Idea: Extrusion Detection
 - Look at what is exiting the network to find attacks
 - Used to detect insider threats and successful intruders

Intrusion Prevention

- ⦿ Intrusion Prevention Systems are IDS that block or drop traffic based on matches
- ⦿ Host based Intrusion Prevention (HIPS)
 - Behavior based (e.g. block buffer overflows)
 - Signature based
- ⦿ Network based Intrusion Prevention (NIPS)
 - When part of a firewall, NIPS is often sold as
 - Unified Threat Management (UTM), or
 - Deep Packet Inspection (DPI)

Drawbacks to IPS

- ⦿ Tuning for false positives reduces their effectiveness
 - If your warehouse alarm system is connected to a machine gun, better make sure it doesn't have a false alarm!
- ⦿ Like a default-allow firewall, tries to enumerate badness
 - In a managed network, the known-good traffic should be manageable, while bad stuff is nearly infinite
- ⦿ Can be useful for preventing well known attacks
 - Think Slammer, Code Red

Drawbacks to I[DP]S

- ⦿ Expanding bandwidth (10GB and beyond)
- ⦿ Encryption (SSL/TLS, IPSec, etc.)
- ⦿ Increased art of evasion
 - Taking advantage of L7 protocols
 - Many types of encoding
 - OS specific fragmentation
 - Operating systems handle reassembly in different ways, which can be exploited to avoid I[DP]S during an attack

NIDS Deployment Options

- ⦿ Network location
 - Outside Firewall
 - Lots of false alarms due to Internet “noise”
 - Inside Firewall
 - Only sees what is allowed through or tries to go out – often the best value intelligence
 - Between internal zones
 - Why is this desktop attacking my file server?
- ⦿ Types of deployment
 - Network taps (copper, optical)
 - Span port
 - Inline

Snort Overview

- ⦿ Open Source (GPL)
- ⦿ Originally written by Marty Roesch
- ⦿ Backed by Sourcefire, Inc.
- ⦿ Snort is an engine
 - Add-ons available for management
 - Alerts
 - Configuration
 - Rules

What Snort Does

- ① Signature and (some) anomaly based detection
- ① Active community contributing signatures
- ① Supports both detection and prevention
 - Prevention with
 - Snort Inline
 - FlexResp/FlexResp2

Building Snort

- ① Install libpcap
- ② `./configure --enable-dynamicplugin`
- ③ `make`
- ④ `sudo make install`

Snort files

- ⦿ /usr/local/bin/snort
- ⦿ /usr/local/etc/snort
 - snort.conf
 - rules/*.rules
- ⦿ /var/log/snort
 - alert
 - snort.log.*

Variables

- ⦿ Variables used by rule authors to determine applicability of rules
- ⦿ Most variables default to any which can greatly increase false positives
- ⦿ Examples:

```
var HOME_NET  
    [10.1.1.0/24,192.168.1.0/24]  
var EXTERNAL_NET !$HOME_NET  
var SMTP_SERVERS  
    [10.1.1.10,192.168.1.50]
```

Preprocessors

- Stream4 – TCP reassembly and state tracking, not granular to the type of host
- Flow – state tracking, portscan detection
- Stream5 (Snort 2.7) – replaces stream4, flow with destination OS based reassembly
- Protocol specific
 - HTTP, Telnet, RPC, DNS, etc.

Output plugins

- Syslog – send to corporate syslog server
- Tcpdump – saves in binary tcpdump format
- Database (depricated) – causes performance problems, especially with a slow or busy database server
- Unified – binary output format
 - Barnyard – reads unified output, inserts into database
- FLoP – Fast Logging Project for Snort
 - Talks to Snort on a Unix socket, buffers output to database so Snort can keep processing packets

Rule Sources

- ◎ Sourcefire feeds
 - Community (free) – subset of total rules
 - Registered (free) – delayed 7 days
 - Direct Feed (\$\$\$) - current
- ◎ Bleeding Edge Threats
 - A community of rule authors, often provides fast response
- ◎ Local Rules
 - Write rules specific to your environment

Rule Management

- ◎ Oinkmaster -
oinkmaster.sourceforge.net
 - Perl script for automatic rule updating
 - Can preserve rule modifications
- ◎ Commercial tools

Rule Anatomy

- protocol source sourceport -> dest destport
- msg: "alert message"
- Match rules
 - Content: "packet content"
- Reference: reference, id
- Classtype: classification
- Sid: unique sid
- Rev: revision

Rule Actions

◎ Standard Actions

- alert – alert and log
- log - log only
- pass - ignore the packet
- activate - alert and activate a dynamic rule
- dynamic – wait until activated by an activate rule, then act as a log rule

◎ Inline Actions

- drop - make iptables drop then log the packet
- reject - make iptables drop the packet, log it, and then send a reset or unreachable message
- sdrop - make iptables drop the packet with no log

Rule Examples

```
alert tcp $TELNET_SERVERS 23 ->
  $EXTERNAL_NET any (msg:"INFO TELNET login
  incorrect"; flow:from_server,established;
  content:"Login incorrect";
  reference:arachnids,127; classtype:bad-
  unknown; sid:718; rev:9;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 23
  (msg:"BLEEDING-EDGE EXPLOIT Solaris
  telnet USER environment vuln Attack
  inbound"; flow:to_server,established;
  content: "|ff fa 27 00 00 55 53 45 52 01
  2d 66|"; rawbytes; classtype:attempted-
  user; reference:url,riosec.com/solaris-
  telnet-0-day;
  reference:url,isc.sans.org/diary.html?n&s
  toryid=2220; sid:2003411; rev:5;)
```

Alert Limiting

⦿ Thresholding

- Limits number of times a rule alerts
- Also global thresholds limit total # of alerts

⦿ Supression

- Useful to keep rule active, but suppress alerts from “problem” hosts or known good sources

```
suppress gen_id 1, sig_id 1852,  
track by_dst, ip 10.1.1.0/24
```

Alert Limiting (Part 2)

⦿ Rule Tuning

- Disable rules that generate false positives (or use suppression)
 - Disabling rules increases performance, lowers memory requirements, risks an attack getting through
- Customize rule locally
 - Use the ! (not operator) in sources or destinations, such as !\$SMTP_SERVERS

⦿ Use Pass Rules

- Pass rules processed before alert, drop, etc,
- Increases overhead

Alert Management Tools

⦿ Open Source

- OSSIM – www.ossim.net
- Sguil – sguil.sourceforge.net
- BASE - base.secureideas.net

⦿ Commercial

- Sourcefire – www.sourcefire.com
- Applied Watch – www.appliedwatch.com

References

- ◎ Snort.Org (home of Snort User's Manual)
 - <http://www.snort.org/>
- ◎ Bleeding Edge Threats
 - <http://www.bleedingthreats.net/>
- ◎ FLoP
 - <http://www.geschke-online.de/FLoP/>
- ◎ Oinkmaster
 - <http://oinkmaster.sourceforge.net/>
- ◎ Phil Wood – MMAP
 - <http://public.lanl.gov/cpw/>
- ◎ The Snort Top 10
 - <http://isc.sans.org/diary.html?storyid=1981>