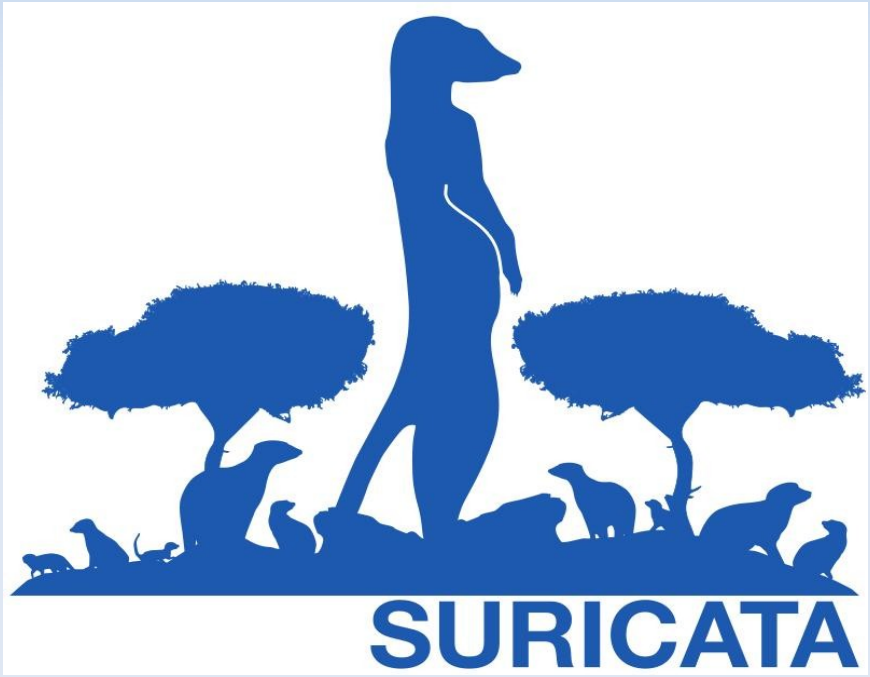
Advances in Suricata



- Eric Leblond
  - @Regiteric
  - http://home.regit.org/
- Victor Julien
  - @inliniac
  - http://www.inliniac.net/

Content of this talk

- Introduction to Suricata, OISF
- Eric Leblond will speak about recent advancements in TLS handling
- I will discuss a new feature: file extraction

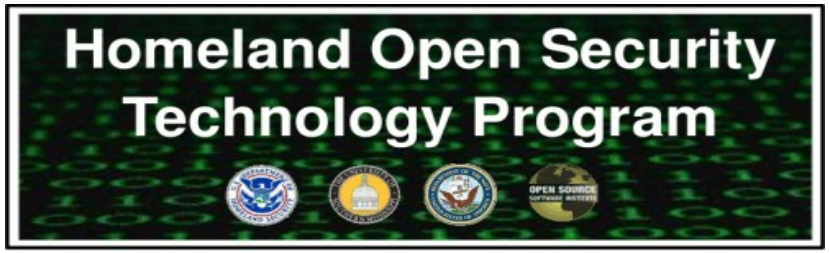
What is Suricata

- Suricata is a Network Intrusion Detection and Prevention System (IDS/IPS)
- Open Source
- Inspects network packets
- (mainly) signature based inspection

Who builds Suricata

- Build by Open Information Security Foundation (OISF)
- US based non-profit
- Funded by DHS
- Supported by consortium of vendors


OISF
OPEN INFORMATION SECURITY FOUNDATION


Homeland Open Security
Technology Program


SURICATA


EMERGING
THREATS PRO
the comprehensive ruleset


bivio
NETWORKS


endace


napatech


BREACH
SECURITY LABS


npulse
TECHNOLOGIES


EVERIS
The Cornerstone of Network Security


nitrosecurity


KERIO


MARA
SYSTEMS


NVIDIA
TESLA
WITH CUDA

How does Suricata IDS work

- placement in the network to see packets
- decoding of packets
- reassembly of IP packets, TCP streams

How does Suricata IDS work (2)

- parsing of higher level protocols (e.g. HTTP)
- detection
- output -- events, alerts

How does Suricata IPS work

- similar to the IDS, however *inline*
- normalization
- blocking

Limitations of an IDS

- easy to overwhelm, packet loss
- impedance mismatch

Example of impedance mismatch

Technology/HTTP back-end	Overall Parsing Result	Example
ASP.NET/IIS	All occurrences of the specific parameter	par1=val1,val2
ASP/IIS	All occurrences of the specific parameter	par1=val1,val2
PHP/Apache	Last occurrence	par1=val2
PHP/Zeus	Last occurrence	par1=val2
JSP,Servlet/Apache Tomcat	First occurrence	par1=val1
JSP,Servlet/Oracle Application Server 10g	First occurrence	par1=val1
JSP,Servlet/Jetty	First occurrence	par1=val1
IBM Lotus Domino	Last occurrence	par1=val2
IBM HTTP Server	First occurrence	par1=val1
mod_perl,libapreq2/Apache	First occurrence	par1=val1
Perl CGI/Apache	First occurrence	par1=val1
mod_perl,lib???/Apache	Becomes an array	ARRAY(0x8b9059c)
mod_wsgi (Python)/Apache	First occurrence	par1=val1
Python/Zope	Becomes an array	['val1', 'val2']
IceWarp	Last occurrence	par1=val2
AXIS 2400	All occurrences of the specific parameter	par1=val1,val2
Linksys Wireless-G PTZ Internet Camera	Last occurrence	par1=val2
Ricoh Aficio 1022 Printer	First occurrence	par1=val1
webcamXP PRO	First occurrence	par1=val1
DBMan	All occurrences of the specific parameter	par1=val1~~val2

Source: http://tacticalwebappsec.blogspot.com/2009/05/http-parameter-pollution.html

Limitations of an IDS (2)

- false positives
- false negatives
- encryption

So what does Suricata do to deal with this a.k.a. Major features

- getting the most out of your hardware: multi threading, hardware capture cards, GPU
- high level protocol detection (HTTP, etc)
- high speed IP matching
- advanced HTTP inspection and logging

Multi-threading

- Multi core is here to stay
- highly modular design of the engine
- scalable

Hardware Capture Card Support

- Endace DAG cards
- Napatech cards (in development)
- PF_RING

GPU acceleration

- CUDA only
- design challenges
- OpenCL?

High level protocol detection

- very helpful in detecting malware
- Previously:

  alert tcp $HOME_NET -> $EXTERNAL_NET $HTTP_PORTS (...detection keywords...)

  $HTTP_PORTS usually set to something like 80:81,8080

High level protocol detection (2)

- Now:

  alert http $HOME_NET -> $EXTERNAL_NET any (...detection keywords...)

  - detection on ANY port

High speed IP matching

- Emerging Threats project has large IP lists of known bad hosts & networks
- You'd like to know if hosts on your network talk to known compromised hosts, don't you?
- Suricata can efficiently load them all and match with low overhead

Advanced HTTP inspection and logging

- HTTP session parsing with libhtp on top of stream reassembly
  – Written by Ivan Ristic of ModSecurity / IronBee fame
- Full HTTP session state reconstruction

Advanced HTTP inspection and logging (2)

- File extraction ... more on that later
- Request logging

HTTP request logging

- normal & extended
- 11/24/2009-18:55:44.663812 192.168.1.42 [**] /x.exe [**] Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1) [**] 192.168.1.1:55868 -> 192.168.1.42:6763
- Extended includes more info, for http_agent

Next up, Eric!


Photograph by Mattias Klum
© 2002 National Geographic Society. All rights reserved.
Meerkats
National Geographic magazine, September 2002

Suricata TLS support

- TLS is an application layer
- Automatic detection
  - Independent of the port
  - Based on pattern matching
- Dedicated keywords
  - Usable in signatures

- Suricata application layer
  - HTTP
  - SMTP
  - FTP
  - SSH
  - DCERPC
  - SMB

A TLS handshake parser

- Handshake parser: No decryption of encrypted traffic
- Method
  - Analysis of TLS handshake
  - Parsing of the TLS messages

- Security oriented parser
  - Code developed from scratch
    - Provide a hackable code-base for the feature
    - No external dependency
    - Contributed by Pierre Chifflier
  - With security in mind
    - Resistance to attack (audited, fuzzed)
    - Anomaly detection

Writing signatures using TLS

- The syntax
  - “alert tcp $HOME_NET any → $EXTERNAL_NET 443”

    Becomes
  - “alert tls $HOME_NET any → $EXTERNAL_NET any”
- Interests
  - No dependency on IP parameters
  - Limit match to the correct protocol
    - Less false positive
    - More performance

TLS keywords

- TLS.version
  - Match on protocol version number
- TLS.subject
  - String match on certificate Subject
- TLS.issuerdn
  - String match on certificate IssuerDN
- More to come

Detecting Rogue certificate

- The conditions
  - Running some servers
  - Having an official PKI
- The sig
  - “alert tls any any → $SERVERS any (tls.issuerdn:!”C=NL, O=Staat der Nederlanden, CN=Staat der Nederlanden Root CA”;)”

Detecting certificate change

- Google.com is signed by Google Internet Authority
  - not diginotar
  - This is bad, let's drop it
- “drop tls $CLIENT any → any any (tls.subject=”C=US, ST=California, L=Mountain View, O=Google Inc, CN=*.google.com”; tls.issuerdn=!”C=US, O=Google Inc, CN=Google Internet Authority”;)”

- What KPN has been hacked too!
  - Let's rock
  - “drop tls $CLIENT any → any any (tls.issuerdn=”C=NL”);”

Current limitation and upcoming evolution

- Match is done on first certificate of the chain
  - Can't do check on chained certificates
  - Parser is compliant, only syntax is missing
- Keywords are missing and will be added
  - Cryptographic algorithm used/proposed
  - Key size
  - Diffie-Hellman parameters
- Statistical study

File extraction

- Currently in development
- Extract files from HTTP sessions: uploads and downloads
- Libmagic used to determine file types
- Powerful rule language extensions

Suricata rule language

- sub set and super set of Snort rule language
- left out old stuff nobody used
- added some new things

Suricata rule language (2)

- Example:

alert tcp $HOME_NET any ->
$EXTERNAL_NET 80 (msg:”example rule”;
content:”EVILSTUFF”; sid:1; rev:1;)

content:”EVILSTUFF”; http_uri; nocase;

File extract rule language extensions

- filemagic

  – alert http any any -> any any (msg:"windows exec"; filemagic:"executable for MS Windows"; sid:1; rev:1;)

- filestore

  – alert http any any -> any any (msg:"windows exec"; filemagic:"executable for MS Windows"; filestore; sid:1; rev:1;)

File extract rule language extensions (2)

- Fileext

  – alert http any any -> any any (msg:"jpg claimed, but not jpg file"; fileext:"jpg"; filemagic:!"JPEG image data"; sid:1; rev:1;)

- Filename

  – alert http any any -> any any (msg:"sensitive file leak"; filename:"secret"; sid:1; rev:1;)

File extract rule language extensions (3)

- Uploads to your webserver that only accepts PDF
  - alert http $EXTERNAL_NET -> $WEBSERVER any (msg:”suspicious upload”; flow:established,to_server; content:”POST”; http_method; content:”/upload.php”; http_uri; filemagic:!"PDF document"; filestore; sid:1; rev:1;)

File extract rule language extensions (4)

- alert http $EXTERNAL_NET -> $WEBSERVER any (msg:”suspicious upload”; flow:established,to_server; content:”POST”; http_method; content:”/upload.php”; http_uri; fileext:!”pdf"; filestore; sid:2; rev:1;)

File extract rule language extensions (5)

- private keys

  alert http $HOME_NET any → $EXTERNAL_NET any (msg:”outgoing private key”; filemagic:”RSA private key”; sid:1; rev:1;)

File extract rule language extensions (6)

- Photoshop and Canon raw files

  drop http $HOME_NET any $EXTERNAL_NET any (msg:”Canon Raw upload”; flow:to_server; filemagic:”Canon CR2 raw image data”; sid:1; rev:1;)

  drop http $HOME_NET any → $EXTERNAL_NET any (msg:”Photoshop upload”; flow:to_server; filemagic:”Adobe Photoshop Image”; sid:2; rev:1;)

File storage

- Each file is stored on disk & accompanied with a meta data file

```
TIME:                   10/02/2009-21:34:53.796083
PCAP PKT NUM:           5678
SRC IP:                 61.191.61.40
DST IP:                 192.168.2.7
PROTO:                  6
SRC PORT:               80
DST PORT:               1091
FILENAME:               /ww/aa5.exe
MAGIC:                  PE32 executable for MS Windows (GUI)
                        Intel 80386 32-bit
STATE:                  CLOSED
SIZE:                   30855
```

- Global limits to storage use

File extract limitations and open issues

- Protocols
- Storage limits
- MS Office files

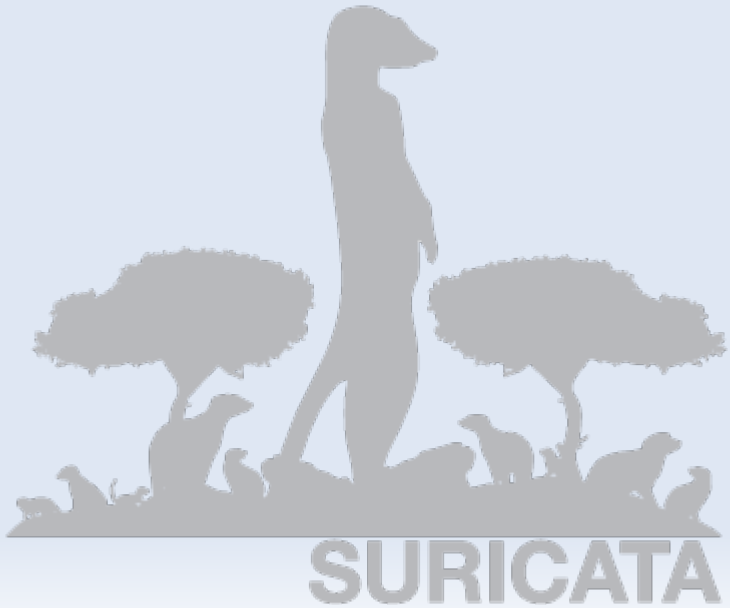
Suricata development

- 2 monthly “stable” release cycle: time based releases
- priorities determined on public brainstorm sessions: last one at RAID 2011, before that RSA San Francisco 2011
- roadmap, bugs, issues in public “redmine” site

Interested in trying Suricata?

- Source
- Debian/Ubuntu/Fedora: old versions
- Security Onion
- Smooth Sec

Thanks for your attention!



OISF
OPEN INFORMATION SECURITY FOUNDATION

Questions?