BRO AND BRO-IDS

Shmoocon 2013

Presented by
Liam Randall
2013-2-17

ABOUT ME

History

- Principal Security Consultant with Giga Co
- 17 Years Consulting (1995)
- BS in CS from XU
- Dozens of Vender Certs
- Speak/Train- Shmoocon, Skydogcon
- “Applied NSM” Summer of 2013

- Bro-IDS
- SecurityOnion

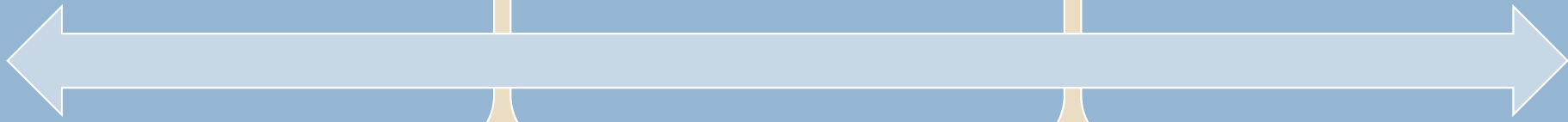
- Liam.Randall@GigaCo.com
- @Hectaman Twitter/IRC

LINKS

Github
github/liamrandall

#Bro_IDS
@Hectaman

bro-ids.org



PRESENTATION OVERVIEW

Bro Basics

- Features
- Network Fit
- Log & Event Structure

Applications

- Standard IDS Cases
  Beyond Signatures
- Advanced Network Discovery
- Complex Traffic Monito
- Brotego: Maltego + Bro

Programming Demo

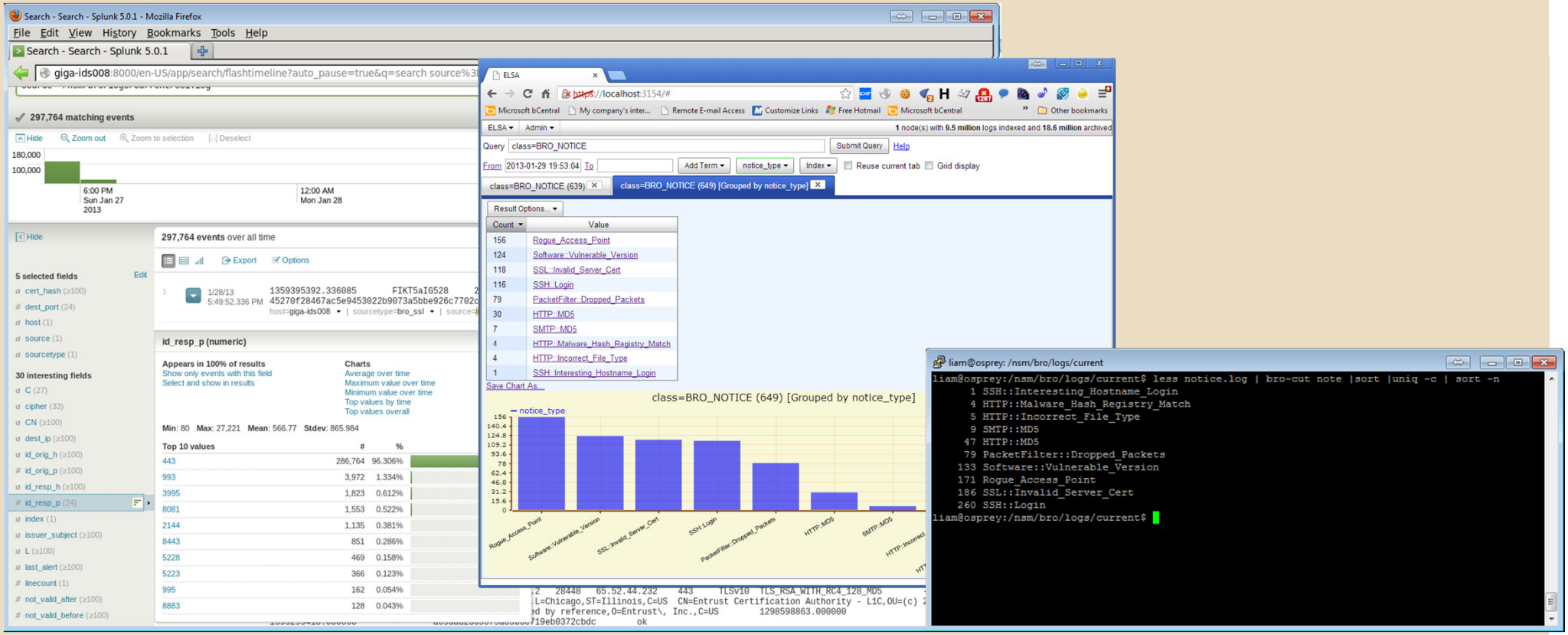
- Custom Scripting
- Lucky 13 Detector!
- HTTP Brute Forcing

WHAT IS BRO

BROGRAMMING

Big Brother

BEGIN WITH THE END IN MIND



BRO PARTS

Types of Bro Data

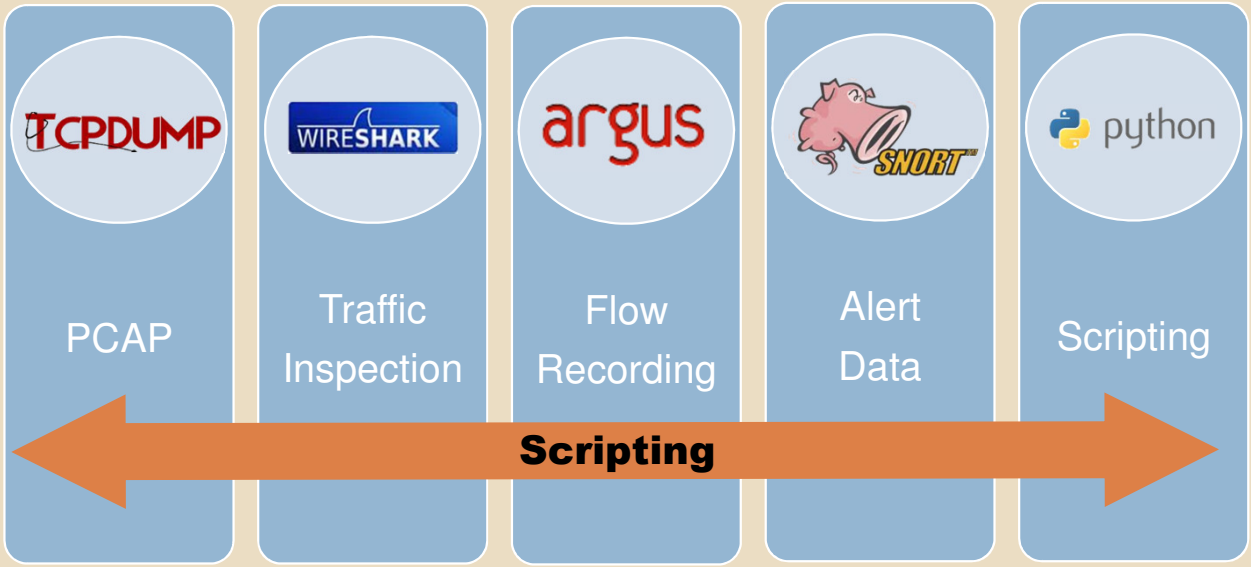
- Signatures
- Logs
- Files
- Traffic
- Pcaps

Interface Methods

- Shell: Bro is Unix-ey
- Splunk
- ELSA

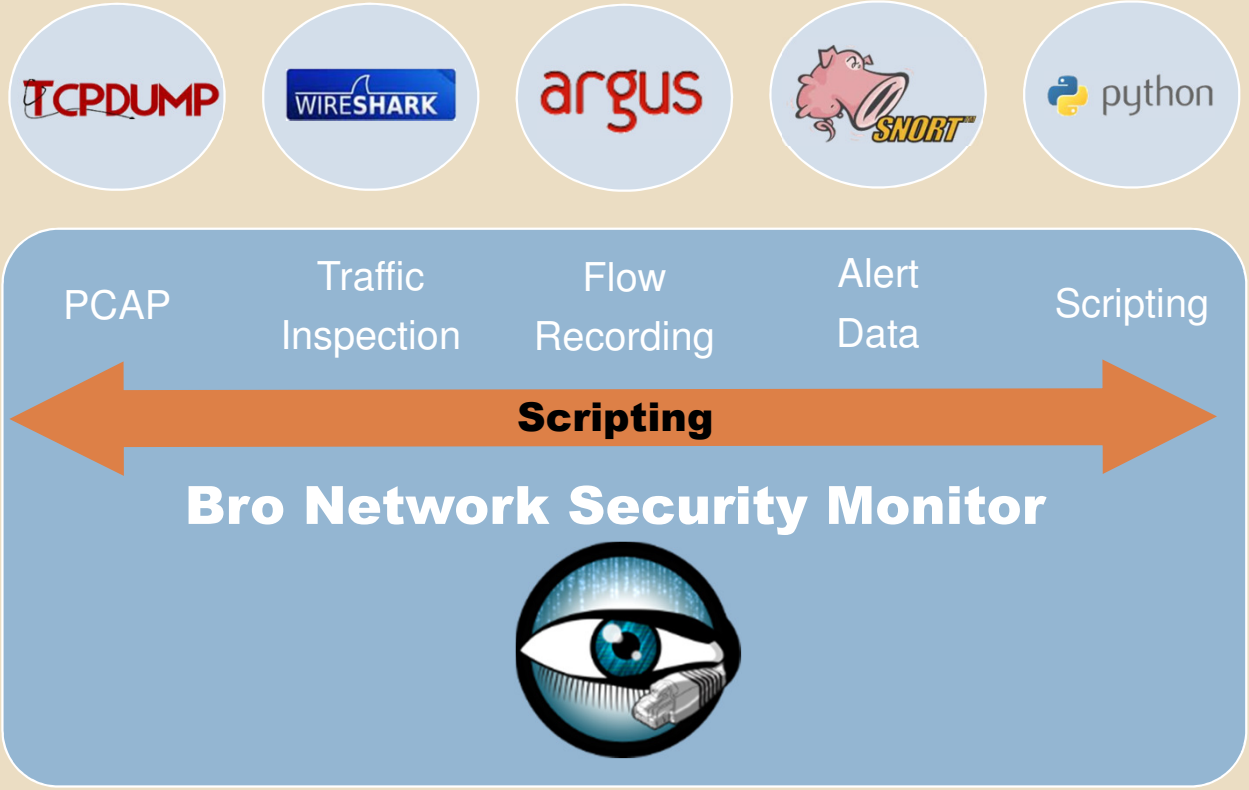
- ArcSight
- Brownian: Elastic Search
- Hadoop
- GNU Parallel (try it!!)
- Google

TRADITIONAL IDS TOOLSET

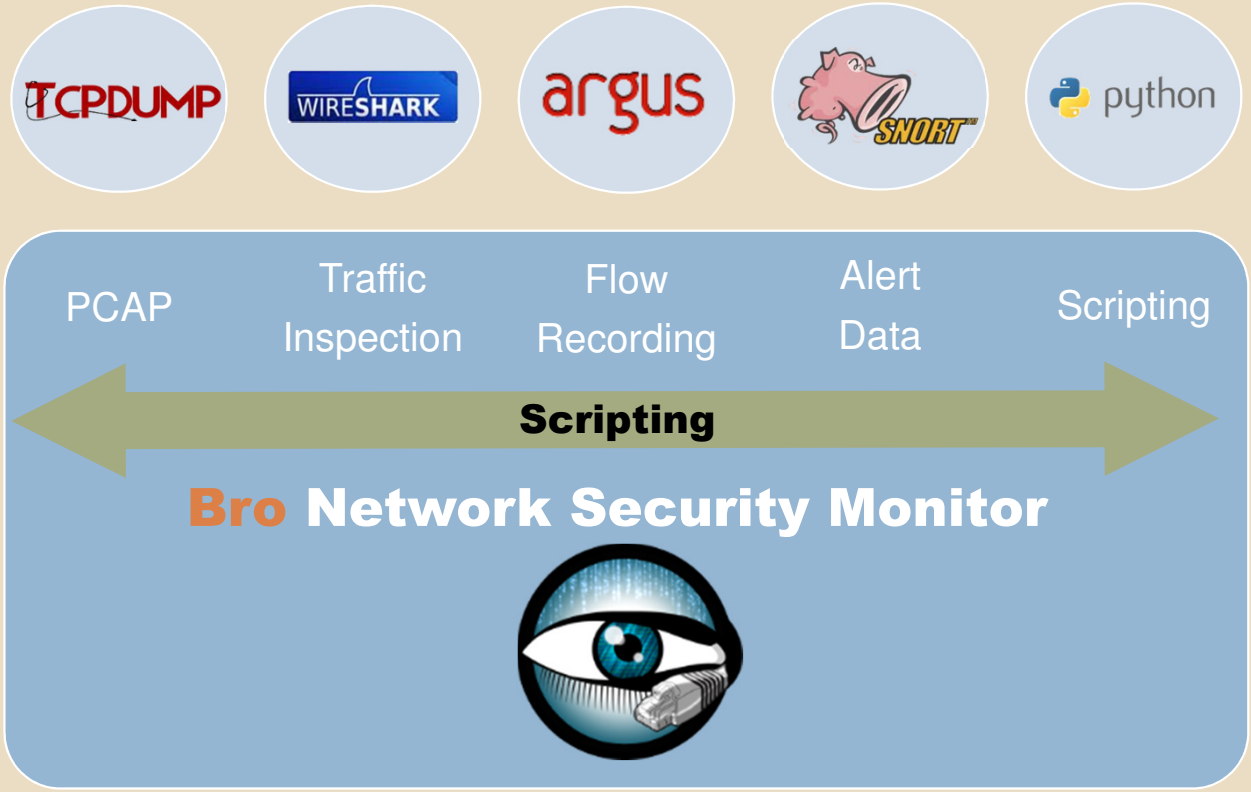


python

TRADITIONAL IDS TOOLSET

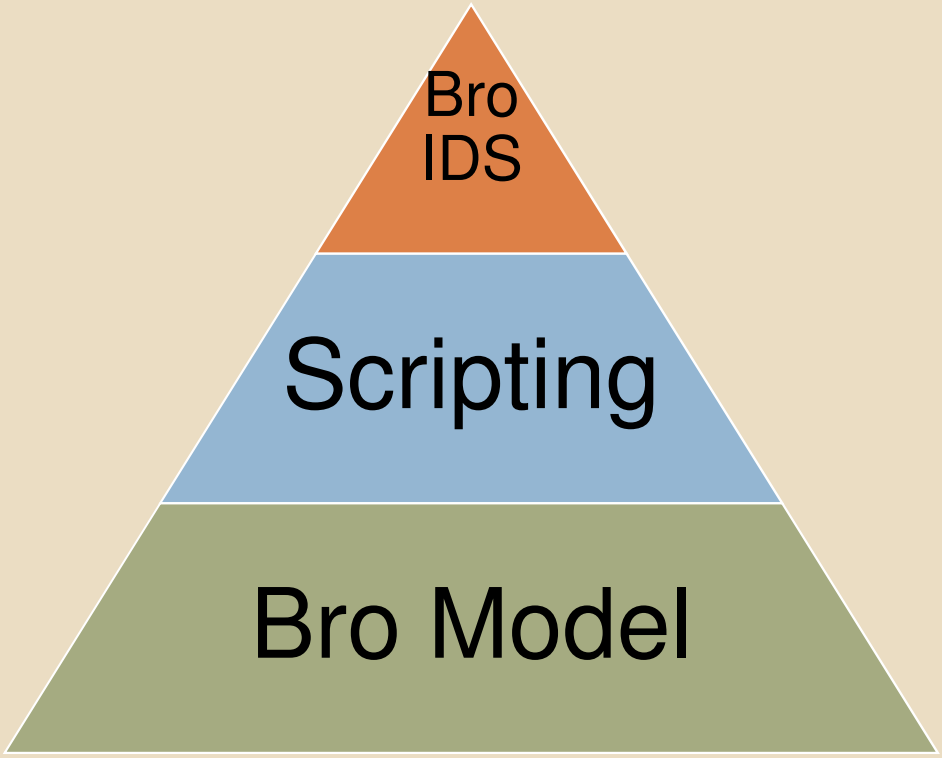


TRADITIONAL IDS TOOLSET



WHAT IS BRO?

Bro Model

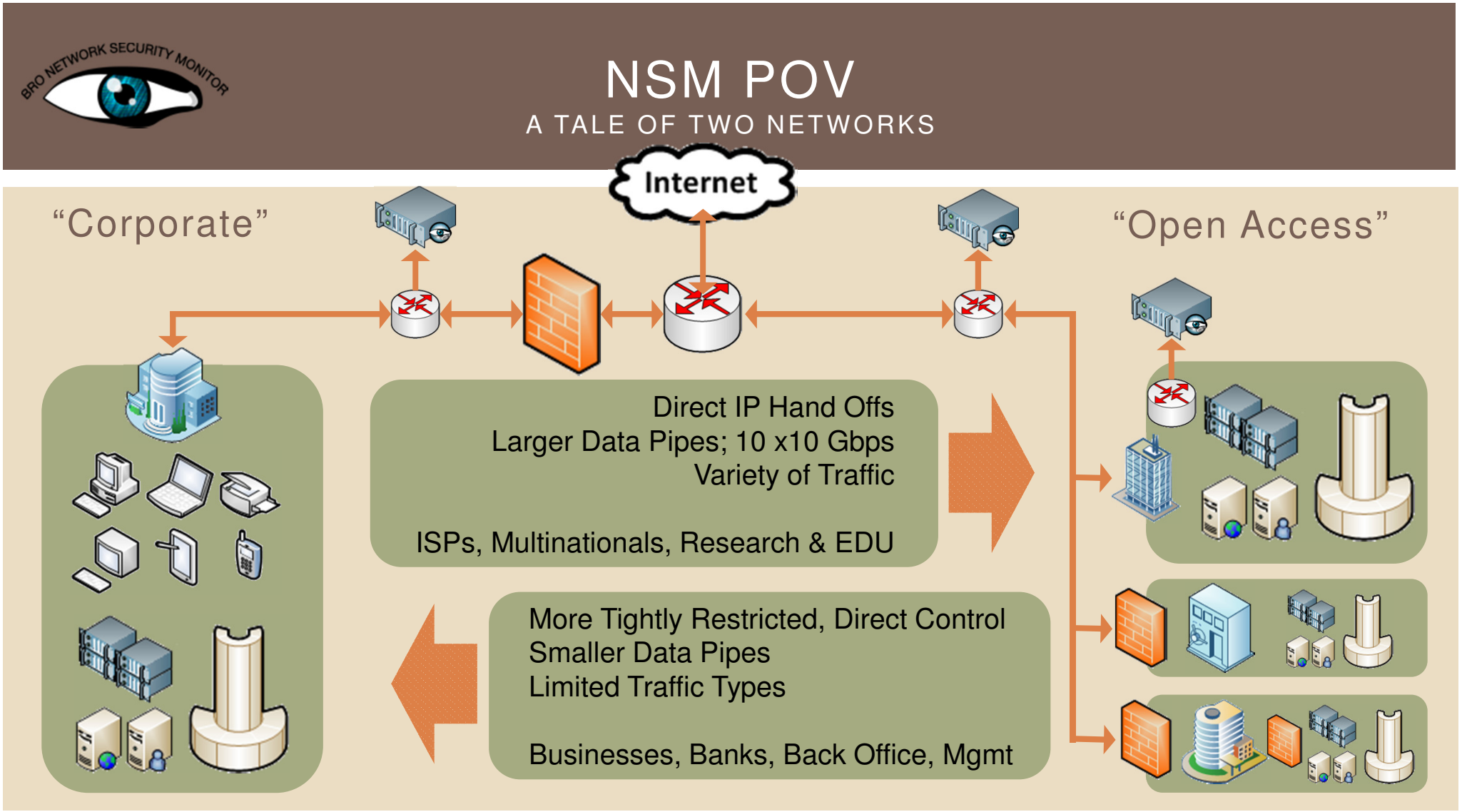


Bro Model

- Not the only way to teach Bro

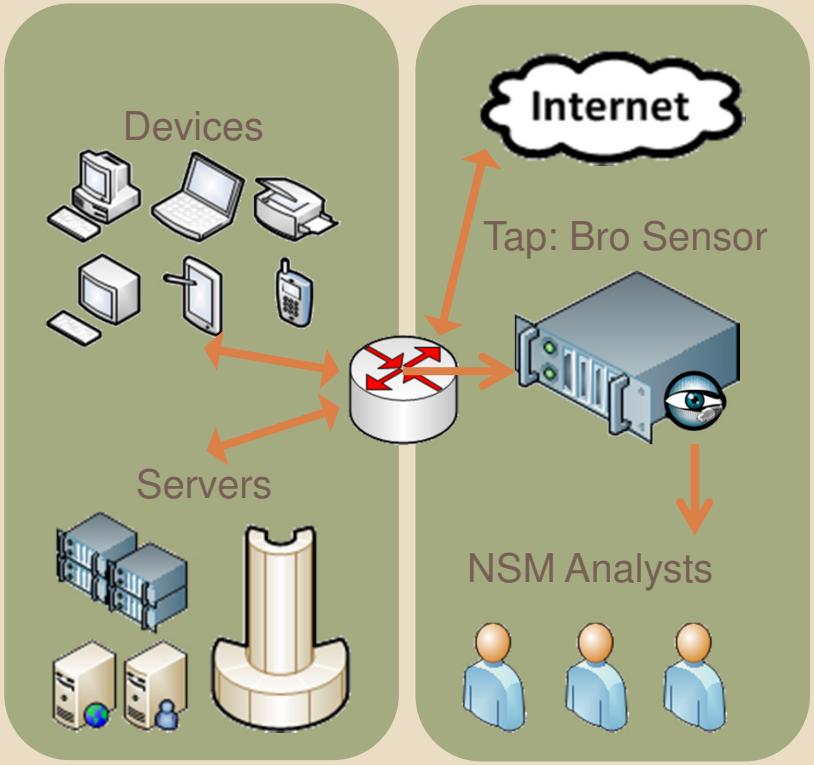
"Bro-IDS is only the first great application to be written in the Bro network programming language."

BRO-IDS


BRO NETWORK SECURITY MONITOR
NSM POV
A TALE OF TWO NETWORKS
Internet
"Corporate"
"Open Access"
Direct IP Hand Offs
Larger Data Pipes; 10 x10 Gbps
Variety of Traffic
ISPs, Multinationals, Research & EDU
More Tightly Restricted, Direct Control
Smaller Data Pipes
Limited Traffic Types
Businesses, Banks, Back Office, Mgmt

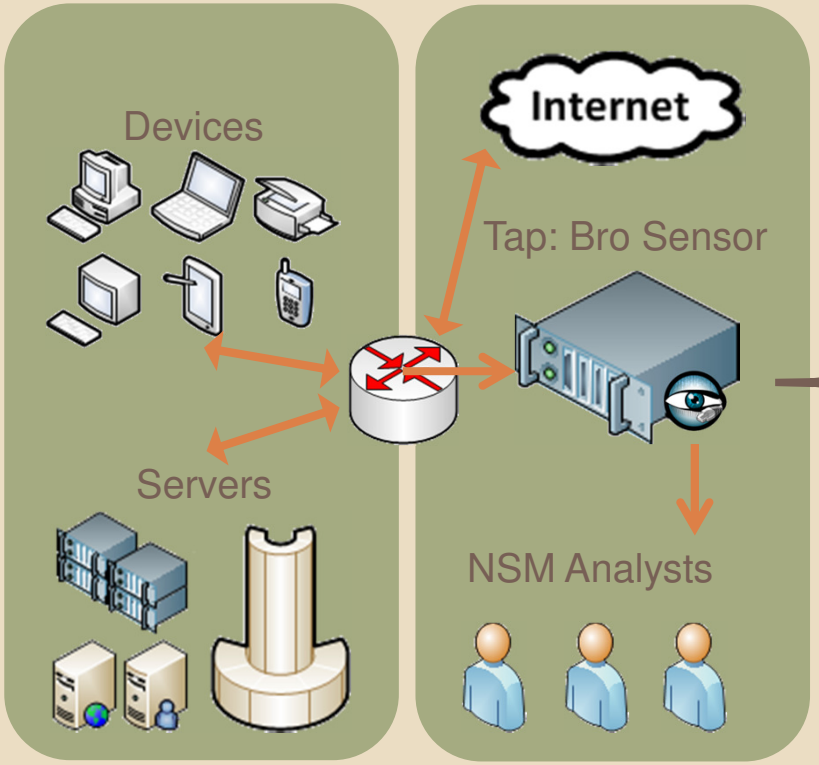
BRO-IDS OVERVIEW

Basic Components



BRO-IDS OVERVIEW

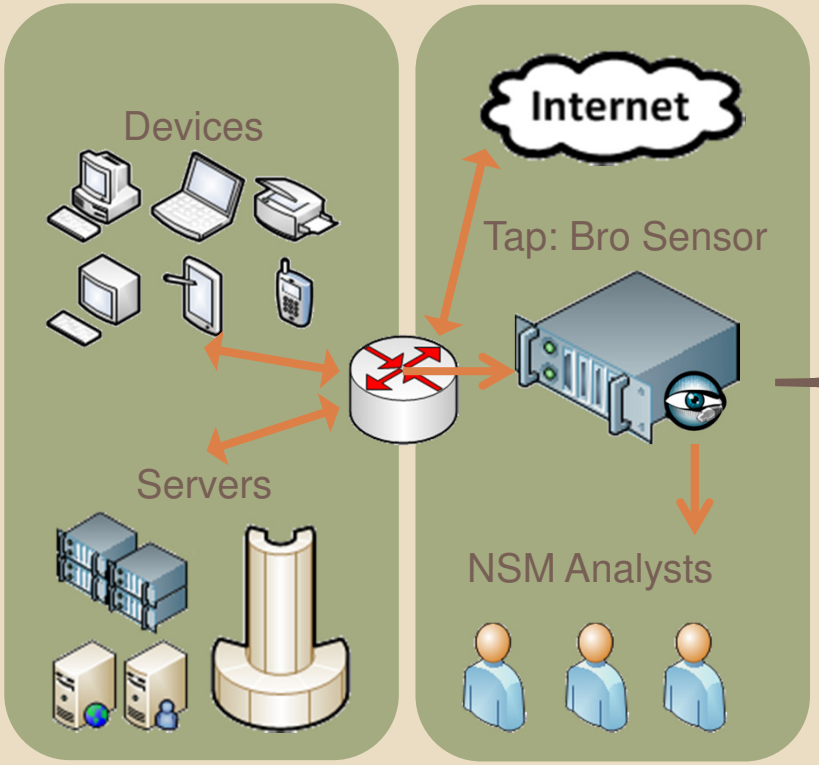
Basic Components



Sensor Analysis Process

BRO-IDS OVERVIEW

Basic Components



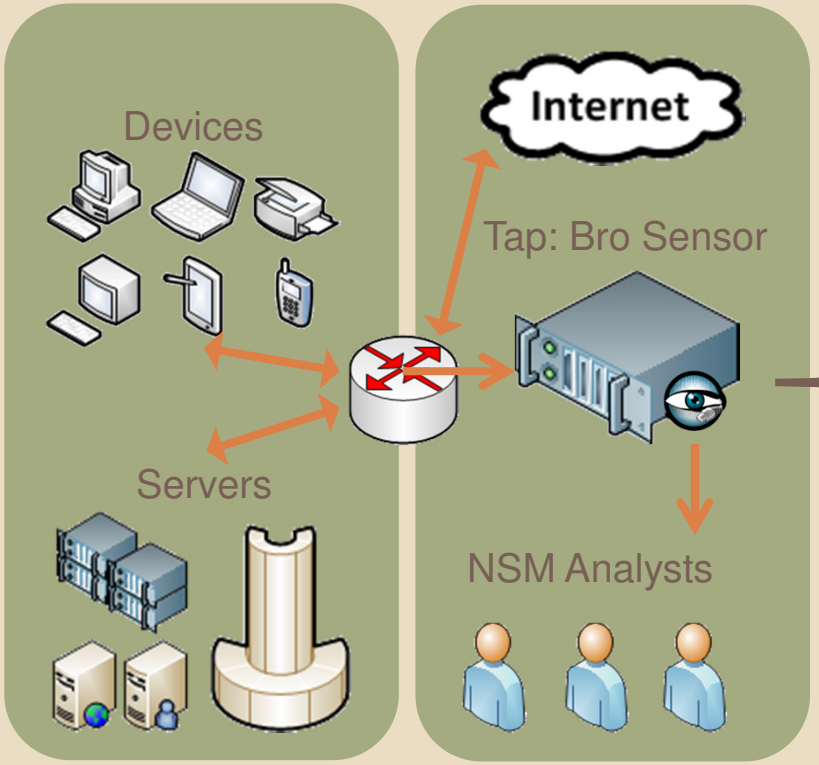
Sensor Analysis Process

Traffic

- Efficient & Flexible Analyzers
- Dynamic Protocol Detection
- Application Layer Semantic Analysis

BRO-IDS OVERVIEW

Basic Components



Sensor Analysis Process

Traffic

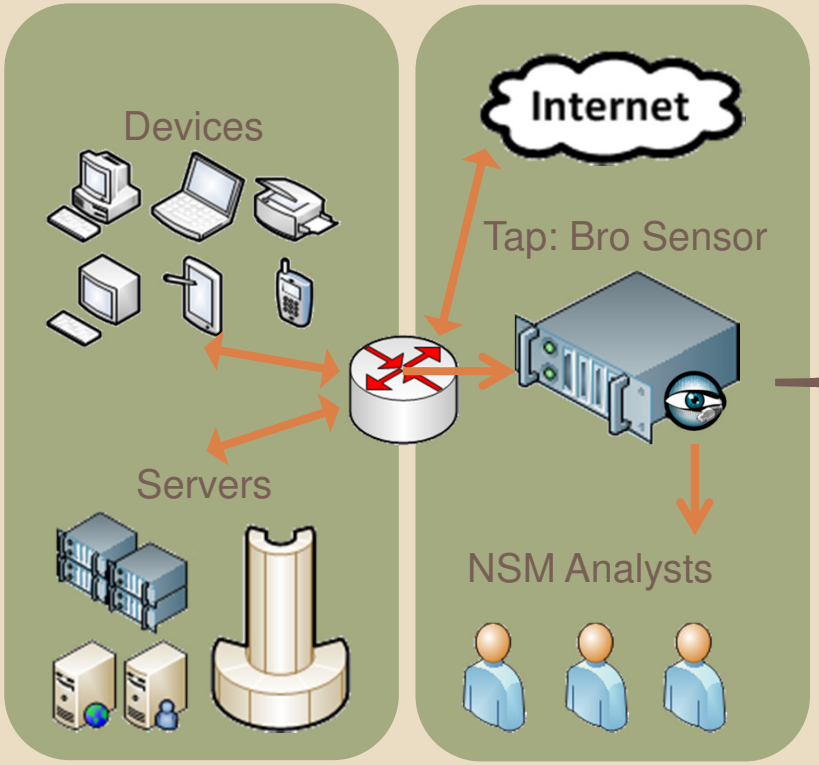
- Efficient & Flexible Analyzers
- Dynamic Protocol Detection
- Application Layer Semantic Analysis

Events

- Highly Stateful, High Performance
- Turing Complete Scripting Language
- Multiple Analysis Frameworks

BRO-IDS OVERVIEW

Basic Components



Sensor Analysis Process

Traffic
- Efficient & Flexible Analyzers
- Dynamic Protocol Detection
- Application Layer Semantic Analysis

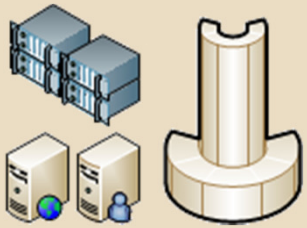
Events
- Highly Stateful, High Performance
- Turing Complete Scripting Language
- Multiple Analysis Frameworks

Structured Output
- High Level Network Archive
- Protocol Specific Detail
- Actions

BRO-IDS
HIGHLY STRUCTURED OUTPUT

ORIGINATORS & RESPONDERS

No CLIENT/SERVER



POV Works

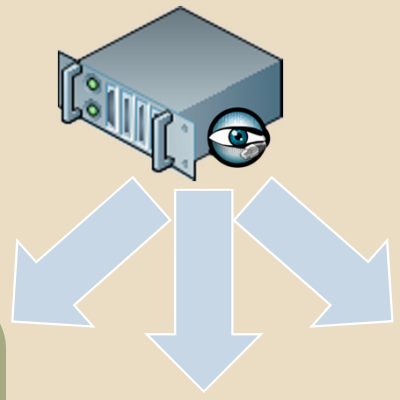
- FTP “up” / “down”
  - Two data channels
  - By byte count?
- SMTP ?

STRUCTURED OUTPUT

Three Classes of Output

- Protocol Logs
  - CONN, HTTP, DNS, FTP, SSL/TLS...
- Actions
  - The Data- attachments, files
  - Act on the Data
  - React to the Data
  - Protocol Specific
  - Turing Complete
- Alerts
  - Notice, Weird → Actions

Model



Protocol

Stateful
Detailed
Configurable

Actions

Turing Complete
Full Spectrum

Alerts

Event Based
Heuristic Based
Pattern Based

1. PROTOCOL: CONN.LOG

- Flow Semantics of TCP/UDP/ICMP Traffic

Ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service
Time	string	addr	port	addr	port	enum	string
1355284742	AZIHpPIejvi	192.168.4.138	68	192.168.4.1	67	udp	-
1326727285	K4xJ9AKH56g	192.168.4.148	55748	196.216.2.3	33117	tcp	ftp-data
1326727283	Jd11tlLtlE	192.168.4.148	58838	196.216.2.3	21	tcp	ftp
1326727287	bVQHYKEz2b4	192.168.4.148	54003	196.216.2.3	31093	tcp	ftp-data
1326727286	5Dki82HwJDk	192.168.4.148	58840	196.216.2.3	21	tcp	ftp
1355284761	YSJ6DDKEzGk	70.199.104.181	8391	192.168.4.20	443	tcp	ssl
1355284791	BqLVVfmVO6d	70.199.104.181	8393	192.168.4.20	443	tcp	ssl
1355284761	ya3SvH6ZxX4	70.199.104.181	8408	192.168.4.20	443	tcp	ssl
1355284812	sxrPWDvcGQ2	192.168.4.20	48433	67.228.181.219	80	tcp	http
1355284903	vlvQgRiHE54	192.168.4.20	14655	192.168.4.1	53	udp	dns
1355284792	gn5FV4jeOJ4	70.199.104.181	8387	192.168.4.20	443	tcp	ssl
1355285010	uEb3j6nYBS7	59.93.52.206	61027	192.168.4.20	25	tcp	smtp
1326962278	SE2LJ7PLwIg	189.77.105.126	3	192.168.4.20	3	icmp	-
1326962279	T6rMQFaMCie	95.165.30.73	3	192.168.4.20	3	icmp	-
1329400936	qtNmAmHhDM4	192.168.4.20	14419	65.23.158.132	6668	tcp	irc
1329400884	cOctAcZusv2	192.168.4.20	32239	89.16.176.16	6666	tcp	irc

1. PROTOCOL: CONN.LOG

- Flow Semantics of TCP/UDP/ICMP Traffic

orig_bytes	resp_bytes	conn_state	local_orig	missed_bytes	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes
count	count	string	bool	count	string	count	count	count	count
-	-	OTH	T	0	C	0	0	0	0
0	59	SF	T	0	ShAdfFa	4	216	4	275
123	323	SF	T	0	ShAdDafF	14	859	12	955
0	145868	SF	T	0	ShAdfFa	68	3544	103	151232
119	324	SF	T	0	ShAdDaFf	15	907	12	956
872	3087	SF	F	0	ShADdFaf	16	1712	11	3671
868	3088	SF	F	0	ShADdFaf	14	1604	11	3672
870	5096	RSTO	F	0	ShADdaFR	14	1594	14	5836
711	421	SF	T	0	ShADadfF	23	1639	24	1677
46	126	SF	T	0	Dd	1	74	1	154
870	5096	RSTO	F	0	ShADdaFR	14	1594	14	5836
804	658	SF	F	0	ShAdDafF	10	1332	11	1242
-	-	OTH	F	0	-	1	159	0	0
-	-	OTH	F	0	-	1	159	0	0
7812	51732	SF	T	0	dDaAFfR	891	43506	1318	104920
15943	276902	SF	T	0	dDaAFf	3334	149403	3698	426622

2. ALERT: NOTICE.LOG

- Classes of data

#fields ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	note
#types	time	string	addr	port	addr	port	enum
1359673187	TLDtWBOrstk	192.168.0.120	61537	50.76.24.57	8443	tcp	SSL::Invalid_Server_Cert
1359673187	L4bDTmPqvs2	192.168.1.8	49540	174.143.119.91	6697	tcp	SSL::Invalid_Server_Cert
1359673187	JAvYksFW1Qb	207.188.131.2	5373	160.109.68.199	8081	tcp	SSL::Invalid_Server_Cert
1359673188	-	192.168.0.57	62220	216.234.192.231	80	tcp	Rogue_Access_Point
1359673188	5OYpDdtlnfd	192.168.0.147	45009	93.174.170.9	443	tcp	SSL::Invalid_Server_Cert
1359673188	-	192.168.0.147	36511	74.125.225.194	80	tcp	Rogue_Access_Point
1359673188	-	-	--	--	--		Software::Vulnerable_Version
1359673188	93CIvevOuxk	192.168.0.147	51897	98.136.223.39	8996	tcp	SSL::Invalid_Server_Cert
1359673209	YpCOvC9p4Ef	208.89.42.50	48620	207.188.131.2	22	tcp	SSH::Login
1359673210	SaKFGzmdXLl	207.188.131.2	11175	23.5.112.107	443	tcp	SSL::Invalid_Server_Cert
1359673214	XLE8fYl5Tvg	207.188.131.2	11677	208.66.139.142	2145	tcp	SSL::Invalid_Server_Cert
1359673214	-	192.168.1.120	60141	74.125.225.195	80	tcp	Rogue_Access_Point
1359673218	NyPHd3qjIKe	208.89.42.50	43891	207.188.131.2	22	tcp	SSH::Login
1359673223	0skn2N4oYbj	192.168.1.116	49249	15.201.49.137	80	tcp	HTTP::MD5
1359673224	Q83ji8AFOO1	192.168.1.116	49250	15.192.45.26	80	tcp	HTTP::MD5
1359673229	WU57HOSwkEj	208.89.42.50	62165	207.188.131.2	22	tcp	SSH::Login

2. ALERT: NOTICE.LOG

msg
enum
SSL certificate validation failed with (self signed certificate)
SSL certificate validation failed with (certificate has expired)
SSL certificate validation failed with (self signed certificate)
Rogue access point detected
SSL certificate validation failed with (certificate has expired)
Rogue access point detected
A vulnerable version of software was detected: Safari 4.0.0-Mobile
SSL certificate validation failed with (unable to get local issuer certificate) emailAddress=qiubz@yahoo-inc.com,CN=android.connector.push.mobile.yahoo.com,OU=PS,O=Yahoo,ST=California,C=US 192.168.0.147
Heuristically detected successful SSH login.
SSL certificate validation failed with (certificate has expired)
SSL certificate validation failed with (unable to get local issuer certificate) emailAddress=keymaster@livevault.com,CN=LiveVault.200345,OU=svc.livevault.com,O=LiveVault Corporation,L=brg009nus,C=US
Rogue access point detected
Heuristically detected successful SSH login.
192.168.1.116 9932c8444e06b32bbb035af5bab31daf http://h19001.www1.hp.com/pub/softpaq/sp57001-57500/sp57398.exe
192.168.1.116 43c32a61aa1fff35dbb450b078c90611 http://h19001.www1.hp.com/pub/softpaq/sp57001-57500/sp57398.exe
Heuristically detected successful SSH login.

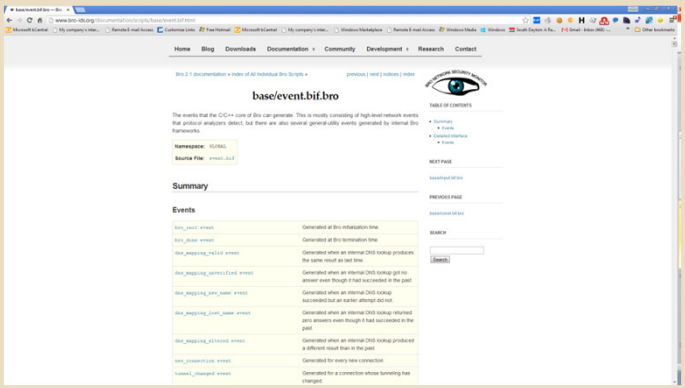
3. ACTIONS

Event Overview

- Bro Model Core
- Bro is network programming language
- Turing Complete
- Events Drive Everything
- Read files, call programs, output data

Documentation

- Events.bif

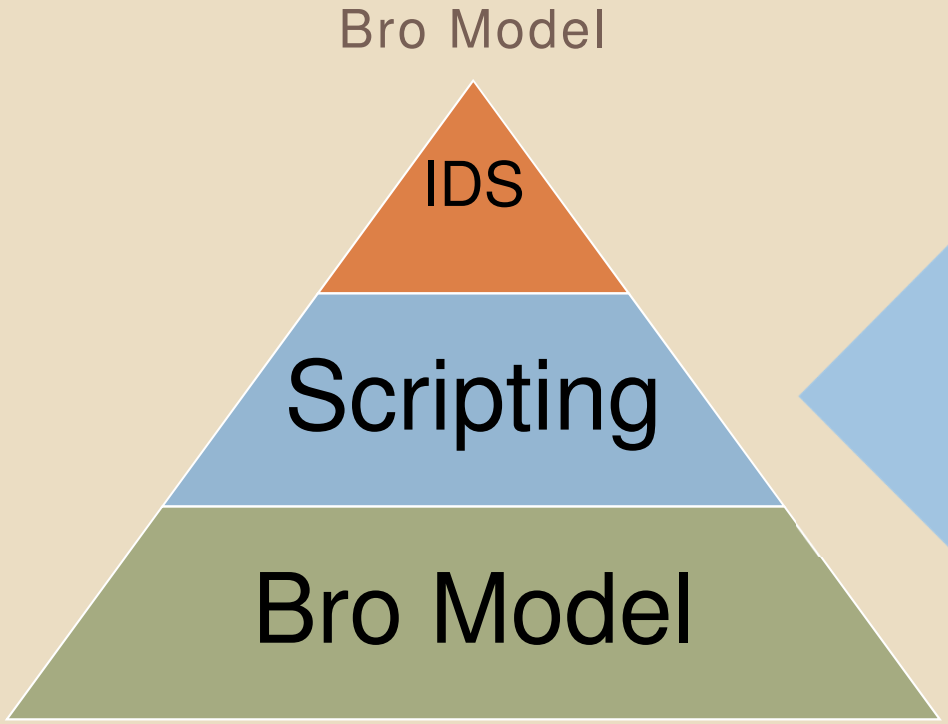


http://www.bro-ids.org/documentation/scripts/base/event.bif.html

BRO

NETWORK PROGRAMMING LANGUAGE EVENTS

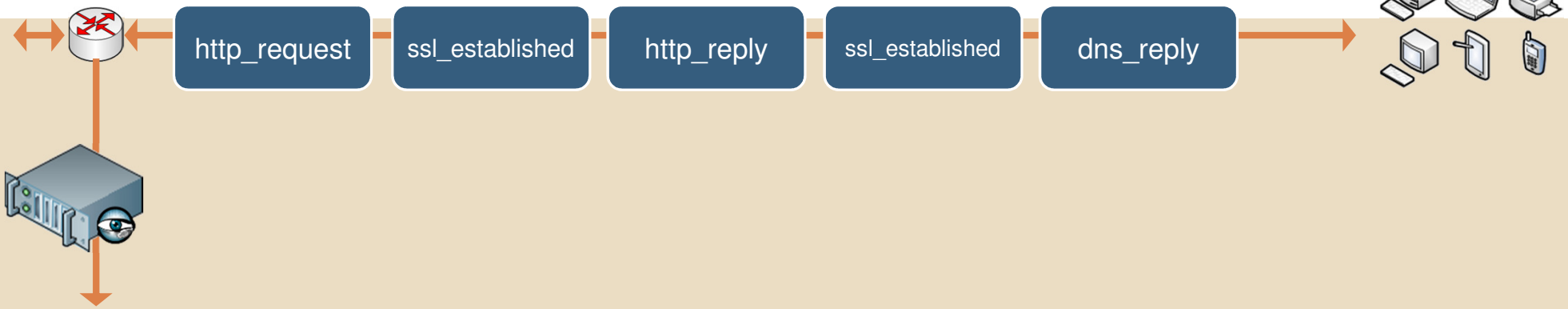
BRO NETWORK PROGRAMMING LANGUAGE



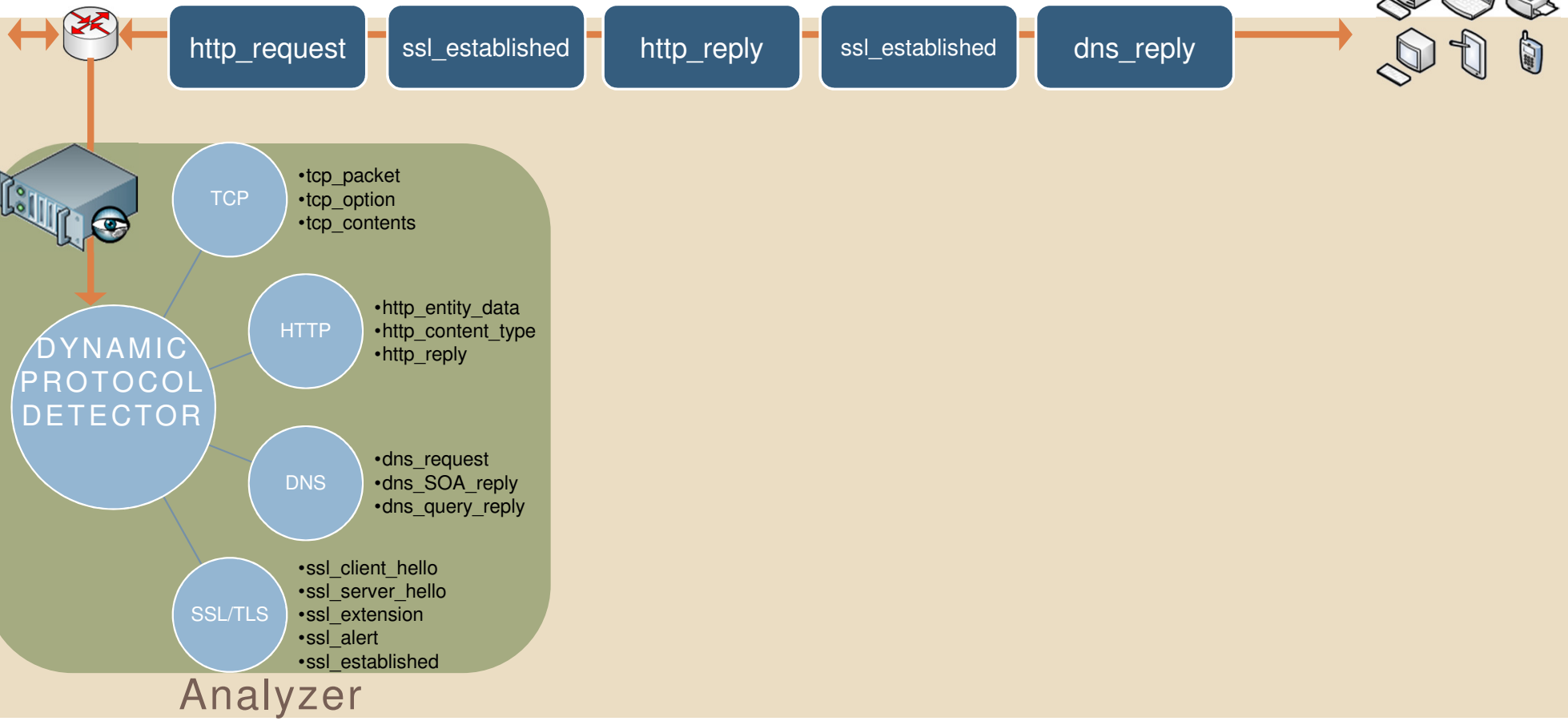
- "Domain Specific Language"
- Vern Paxson, author of Flex lexical analyzer
- Bro is old → 1994

You are here.

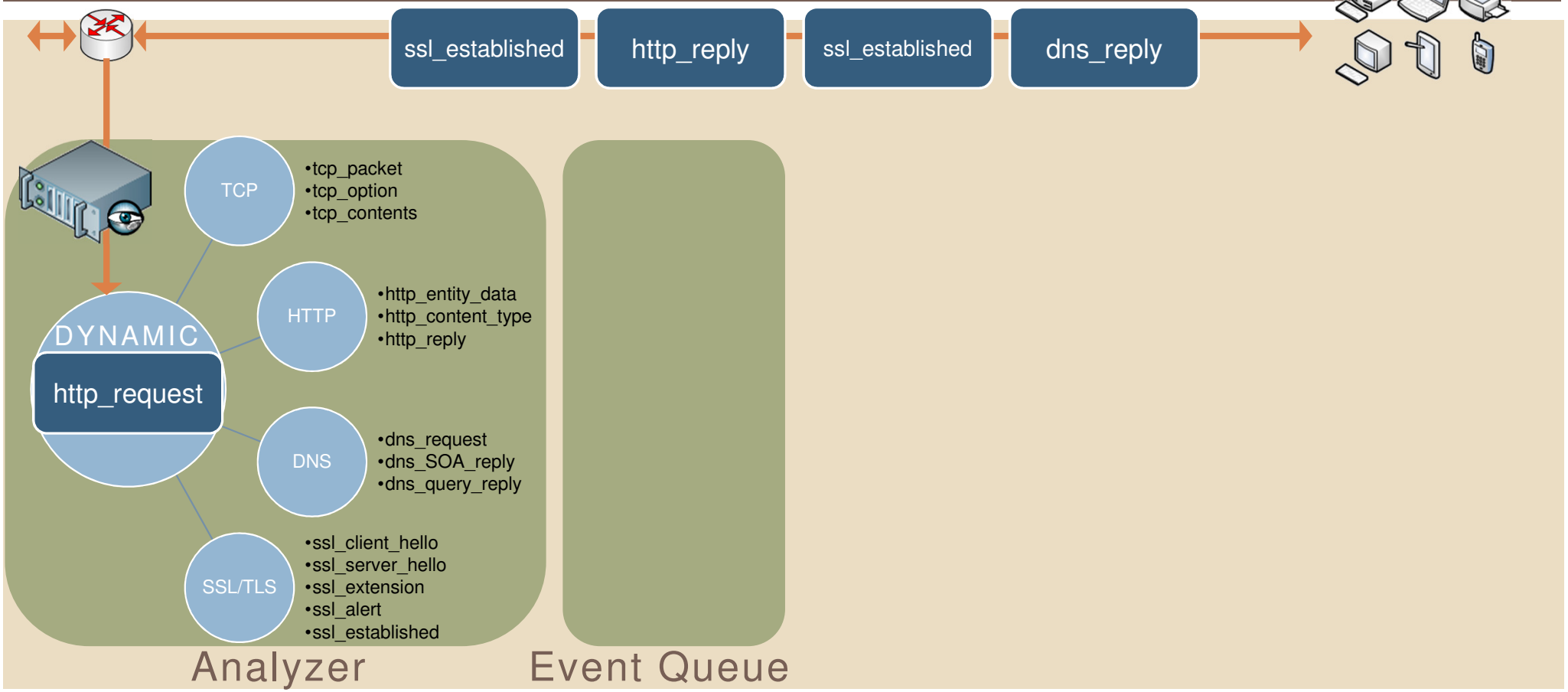
BRO EVENT QUEUE



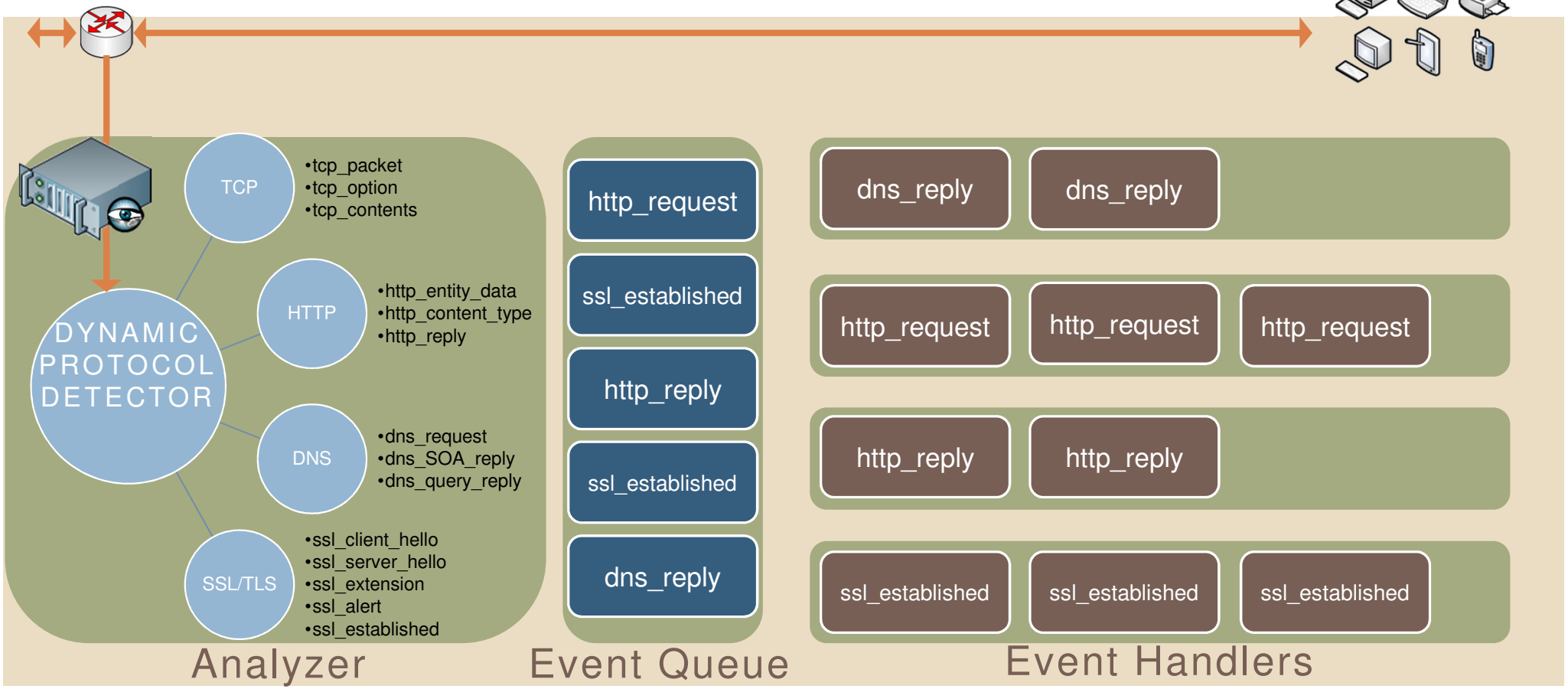
BRO EVENT QUEUE



BRO EVENT QUEUE

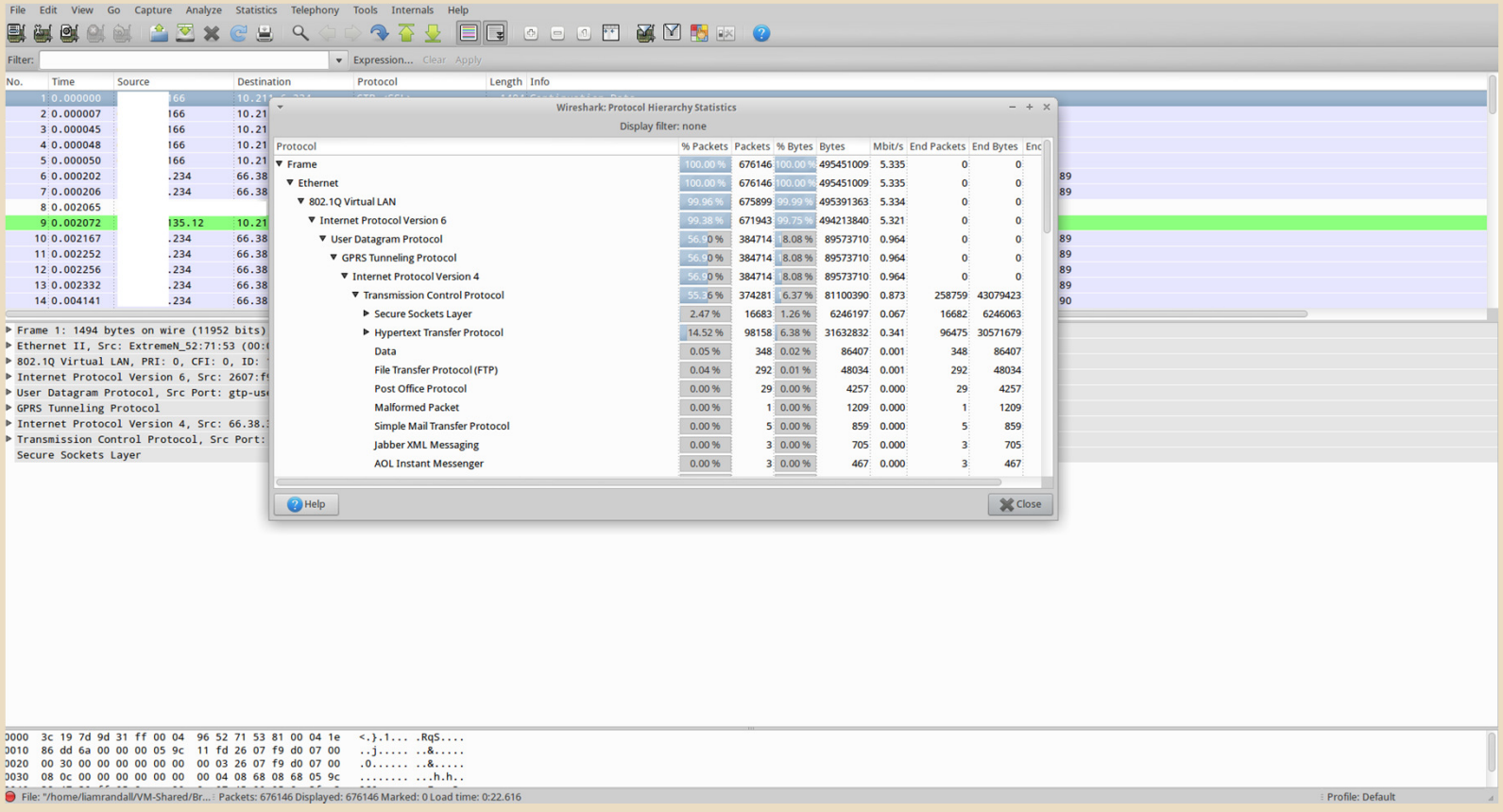


BRO EVENT QUEUE




BRO NETWORK SECURITY MONITOR

BROCEPTION


File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Length Info
Wireshark: Protocol Hierarchy Statistics
Display filter: none
Protocol % Packets Packets % Bytes Bytes Mbit/s End Packets End Bytes
Frame 100.00 % 676146 100.00 % 495451009 5.335 0 0
Ethernet 100.00 % 676146 100.00 % 495451009 5.335 0 0
802.1Q Virtual LAN 99.96 % 675899 99.99 % 495391363 5.334 0 0
Internet Protocol Version 6 99.38 % 671943 99.75 % 494213840 5.321 0 0
User Datagram Protocol 56.90 % 384714 8.08 % 89573710 0.964 0 0
GPRS Tunneling Protocol 56.90 % 384714 8.08 % 89573710 0.964 0 0
Internet Protocol Version 4 56.90 % 384714 8.08 % 89573710 0.964 0 0
Transmission Control Protocol 55.36 % 374281 6.37 % 81100390 0.873 258759 43079423
Secure Sockets Layer 2.47 % 16683 1.26 % 6246197 0.067 16682 6246063
Hypertext Transfer Protocol 14.52 % 98158 6.38 % 31632832 0.341 96475 30571679
Data 0.05 % 348 0.02 % 86407 0.001 348 86407
File Transfer Protocol (FTP) 0.04 % 292 0.01 % 48034 0.001 292 48034
Post Office Protocol 0.00 % 29 0.00 % 4257 0.000 29 4257
Malformed Packet 0.00 % 1 0.00 % 1209 0.000 1 1209
Simple Mail Transfer Protocol 0.00 % 5 0.00 % 859 0.000 5 859
Jabber XML Messaging 0.00 % 3 0.00 % 705 0.000 3 705
AOL Instant Messenger 0.00 % 3 0.00 % 467 0.000 3 467
Help
Close
Frame 1: 1494 bytes on wire (11952 bits)
Ethernet II, Src: ExtremeN_52:71:53 (00:
802.1Q Virtual LAN, PRI: 0, CFI: 0, ID:
Internet Protocol Version 6, Src: 2607:f
User Datagram Protocol, Src Port: gtp-us
GPRS Tunneling Protocol
Internet Protocol Version 4, Src: 66.38.
Transmission Control Protocol, Src Port:
Secure Sockets Layer
0000 3c 19 7d 9d 31 ff 00 04 96 52 71 53 81 00 04 1e <.}.1... .RqS....
0010 86 dd 6a 00 00 00 05 9c 11 fd 26 07 f9 d0 07 00 ..j..... ..&.....
0020 00 30 00 00 00 00 00 00 00 03 26 07 f9 d0 07 00 .0...... ..&.....
0030 08 0c 00 00 00 00 00 00 00 04 08 68 08 68 05 9c ....h.h..
File: "/home/liamrandall/VM-Shared/Br... Packets: 676146 Displayed: 676146 Marked: 0 Load time: 0:22.616
Profile: Default

SSL V2 DETECTOR

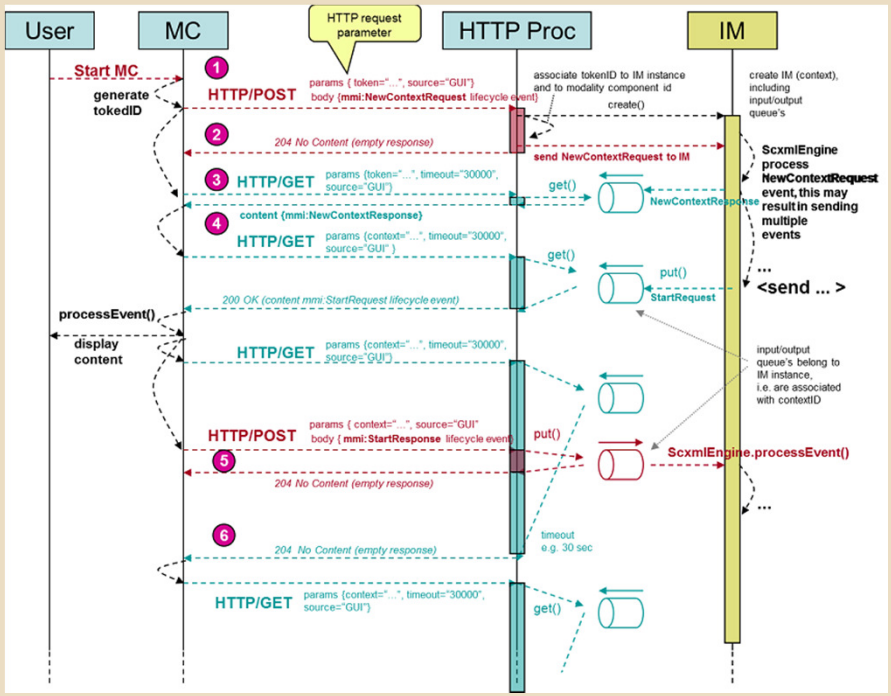
```
event ssl_client_hello(c: connection, version: count, possible_ts: time, session_id:
string, ciphers: count_set)
        {
                if ( version == SSLv2 )
                                {
                                local message = fmt("SSL client %s sent v2 hello",
        c$id$orig_h);
                                local ident = fmt("%s", c$id$orig_h);
                                NOTICE([$note=SSLv2_Client_Hello,
                                                $msg=message,
                                                $conn=c, $identifier=ident]);
                                }
        }
```

EVENT SAMPLE

HTTP Events

http_request: event &group = "http-request"	Generated for HTTP requests.
http_reply: event &group = "http-reply"	Generated for HTTP replies.
http_header: event &group = "http-header"	Generated for HTTP headers.
http_all_headers: event &group = "http-header"	Generated for HTTP headers, passing on all headers of an HTTP message at once.
http_begin_entity: event &group = "http-body"	Generated when starting to parse an HTTP body entity.
http_end_entity: event &group = "http-body"	Generated when finishing parsing an HTTP body entity.
http_entity_data: event &group = "http-body"	Generated when parsing an HTTP body entity, passing on the data.
http_content_type: event &group = "http-body"	Generated for reporting an HTTP body's content type.
http_message_done: event &group = "http-body"	Generated once at the end of parsing an HTTP message.
http_event: event	Generated for errors found when decoding HTTP requests or replies.
http_stats: event	Generated at the end of an HTTP session to report statistics about it.

HTTP State Diagram

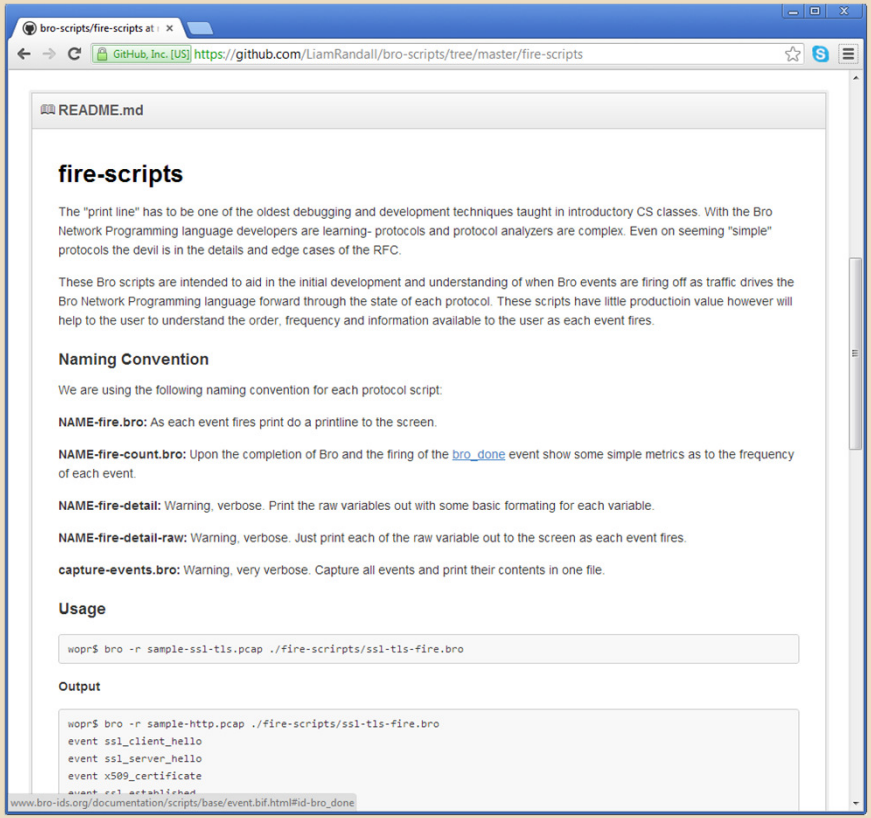


HTTP State Diagram courtesy of w3.org http://www.w3.org/TR/mmi-arch/Images/HTTP_lifecycle_transport_4.png

FIRE-SCRIPTS

HTTP Events

http_request: event &group = "http-request"	Generated for HTTP requests.
http_reply: event &group = "http-reply"	Generated for HTTP replies.
http_header: event &group = "http-header"	Generated for HTTP headers.
http_all_headers: event &group = "http-header"	Generated for HTTP headers, passing on all headers of an HTTP message at once.
http_begin_entity: event &group = "http-body"	Generated when starting to parse an HTTP body entity.
http_end_entity: event &group = "http-body"	Generated when finishing parsing an HTTP body entity.
http_entity_data: event &group = "http-body"	Generated when parsing an HTTP body entity, passing on the data.
http_content_type: event &group = "http-body"	Generated for reporting an HTTP body's content type.
http_message_done: event &group = "http-body"	Generated once at the end of parsing an HTTP message.
http_event: event	Generated for errors found when decoding HTTP requests or replies.
http_stats: event	Generated at the end of an HTTP session to report statistics about it.



HTTP State Diagram courtesy of w3.org http://www.w3.org/TR/mmi-arch/Images/HTTP_lifecycle_transport_4.png

THE BRO-IDS EFFECT

ACTIVE NETWORK MANAGEMENT

ACTIVE NETWORK MANAGEMENT

Enforcement Metrics

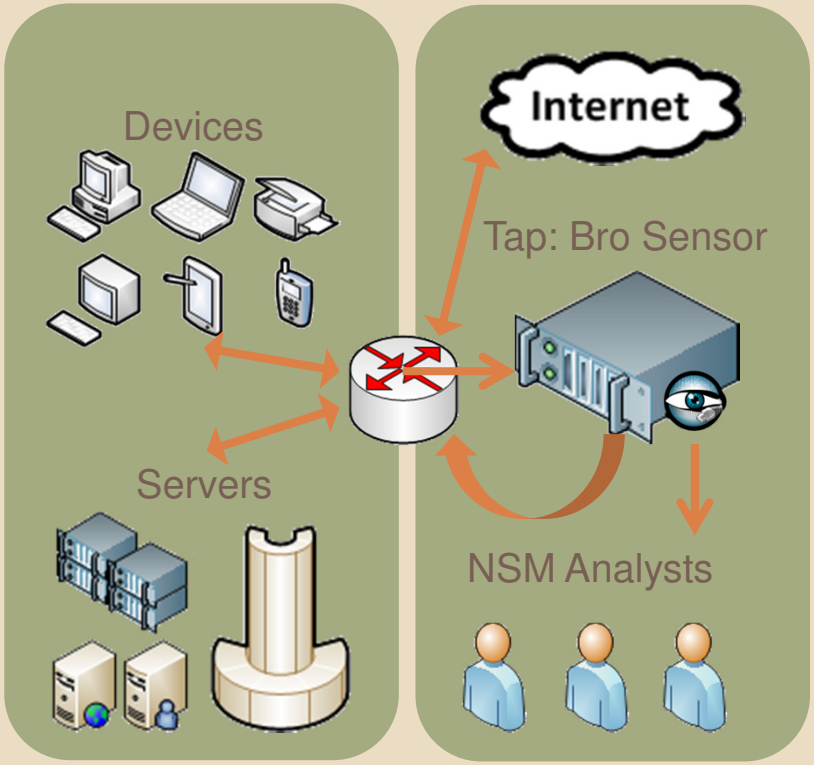
- Software Versions
- Browser Plugin Versions
- Remediation Status
- Real Time Activity
  - If Intel hit & successful EXE download
  - If user agent = Java
- Behavioral

Enforcement Methodologies

- NAC
- Remediation VLAN
- Block Internet Access
- Email on Detection

ACTIVE NETWORK MANAGEMENT

Basic Components

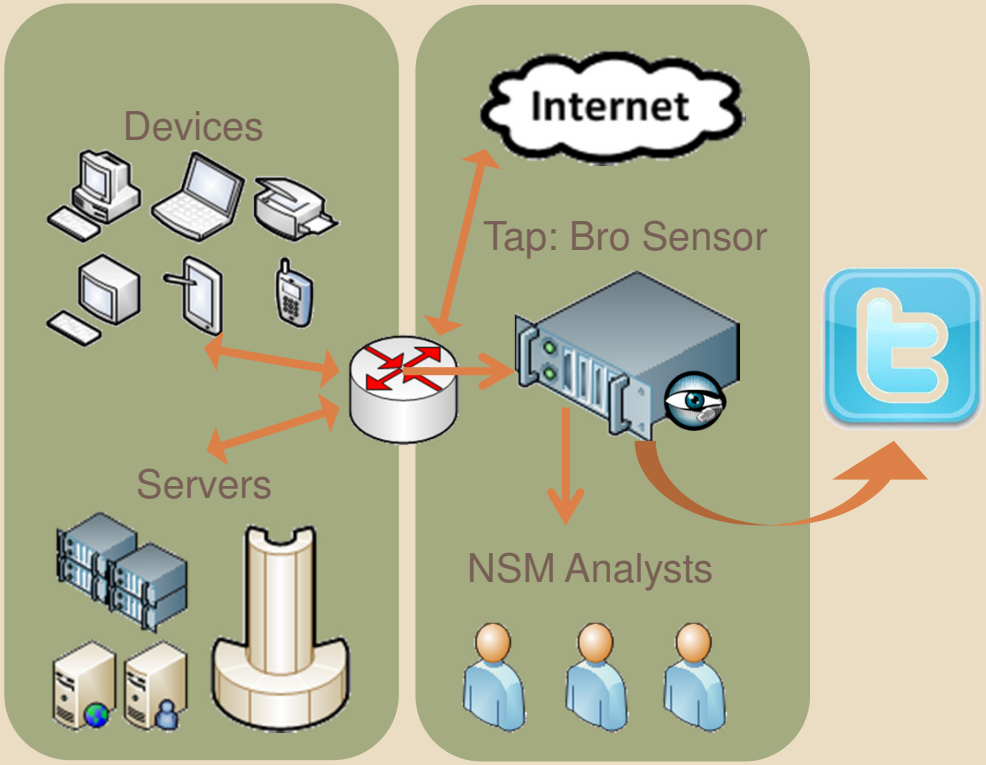


Catch and Release

- Global Intelligence
  - Attacked in Wichita? Secure Everywhere.
- Google Caprica
  - Multiplatform Mgmt- Cisco, Juniper..
- Focus Updates
- Detect & Deny
  - Vendors, Appliances

TWITTER: ACTIVE NETWORK

Basic Components



Catch and Release

- Detect Something
- Do something
- Do something else

....

- Demo
- Profit ?

INTELLIGENCE FEEDS

APPLIED INTEL

Types

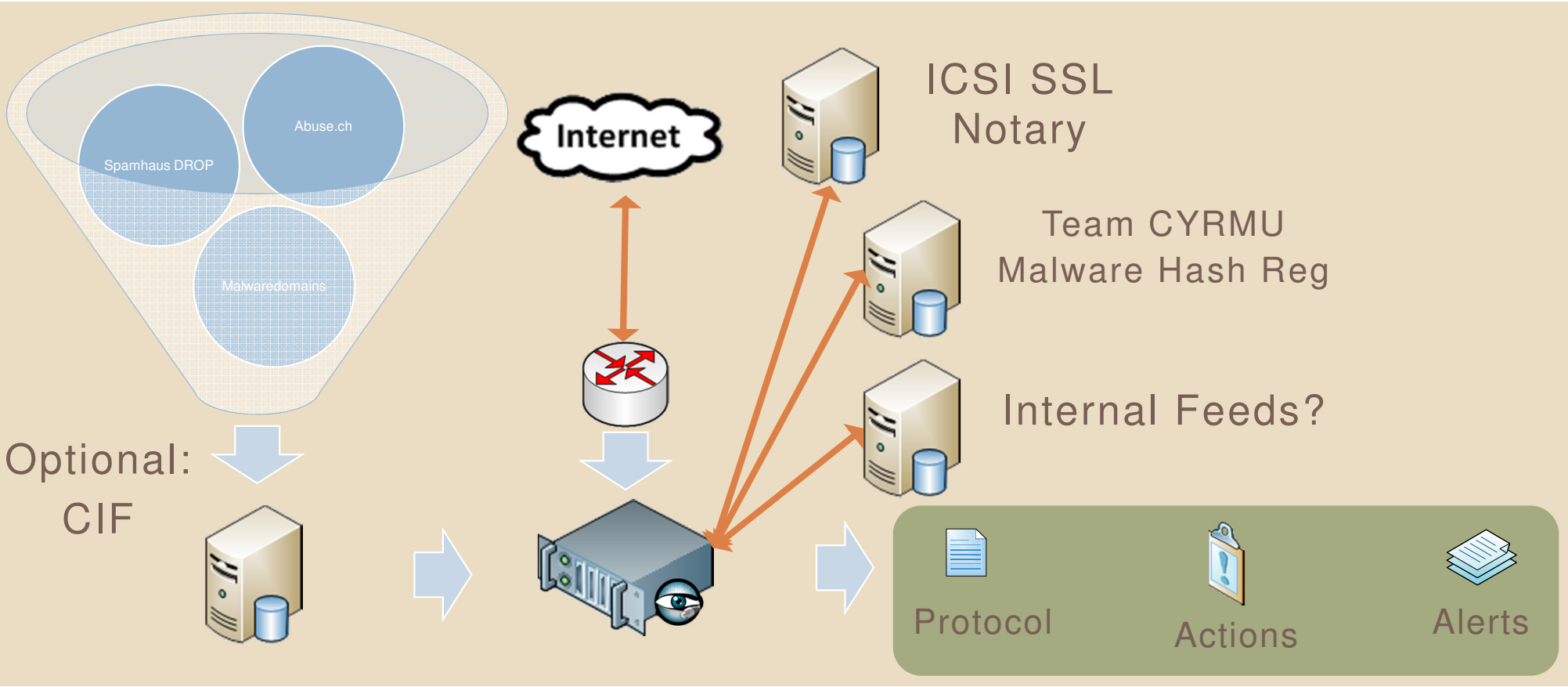
- Passive Intelligence
  - DNS Names
  - IPv4 / IPv6 Addresses
  - Geospatial
  - URL
  - Hash- MD5, SHA1
- Active Intelligence- DNS Based
  - Team CYMRU
  - ICSI SSL Notary

Protocol Monitoring

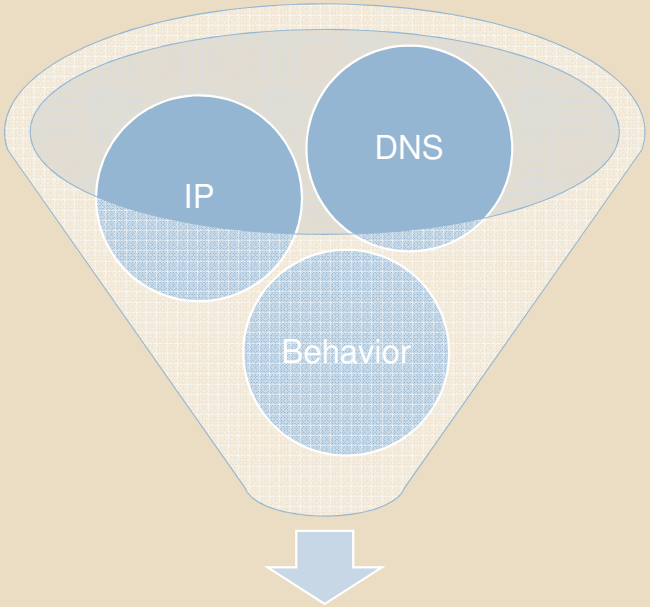
- HTTP, HTTPS
- FTP
- SSL Certs
- SSH
- VPN
- DNS

- → Connections

INTEL OVERVIEW



LIAMS LAW



Signatures

"For every signature hacking away at the leaves of evil there is a greater heuristic striking at its root."

FILE EXTRACTION

FILE EXTRACTION OVERVIEW

Per Protocol Settings

- Multiple Extraction Criteria
  - Geo Spatial→ Country of Origin
  - Signature Based
  - Destination Based- IP, Recipient
- FTP, HTTP, SMTP, IRC
- Other Analyzer in the Works
  - Bittorrent, SMB…
- File Framework in Bro-IDS 2.2

Examples

FTP:

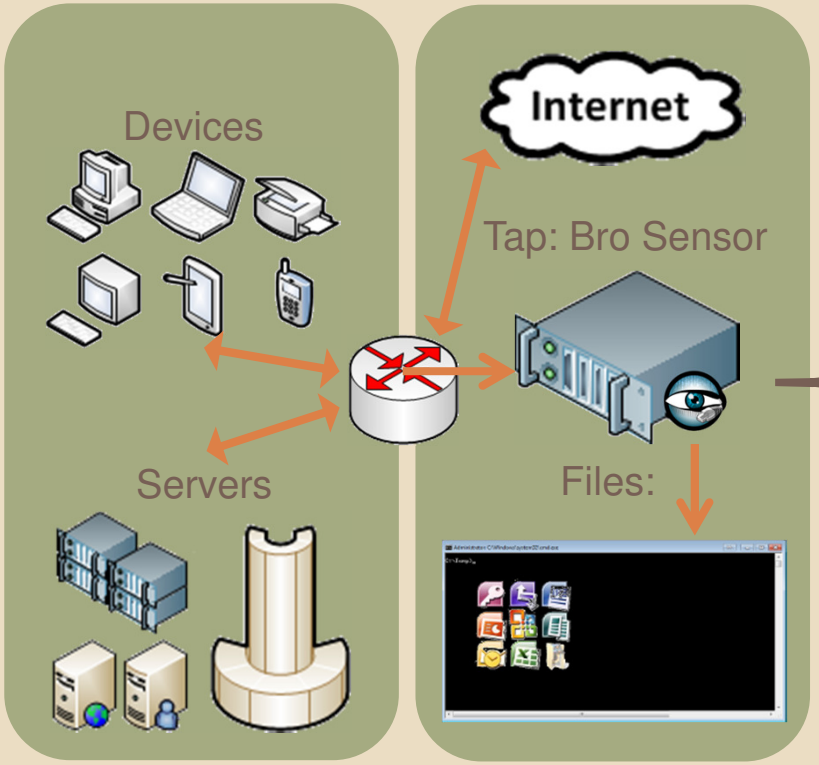
```
const extract_file_types =
      /application\/octet-stream/    |
      /text\/plain/                  |
      /application\/x-dosexec/
      &redef;
```

HTTP:

```
const extract_file_types =
      /application\/x-dosexec/       |
      /application\/x-executable/
      &redef;
```

FILE ANALYSIS

Sensor Components



Extracted File Analysis

Signature Analysis

- Active Analysis→ Malware Hash Registry
- Intel Comparison → OSINT, FS-ISAC, DOE CIRC…

Active Analysis

- www.Malware-Tracker.com
  - Static & Dynamic Analysis
- Cuckoo Box? Volatility

Long Term Analysis

- Coverage for Mobile Devices, Embedded
- Post Compromise Research
- Analysis- copy of every EXE in Company

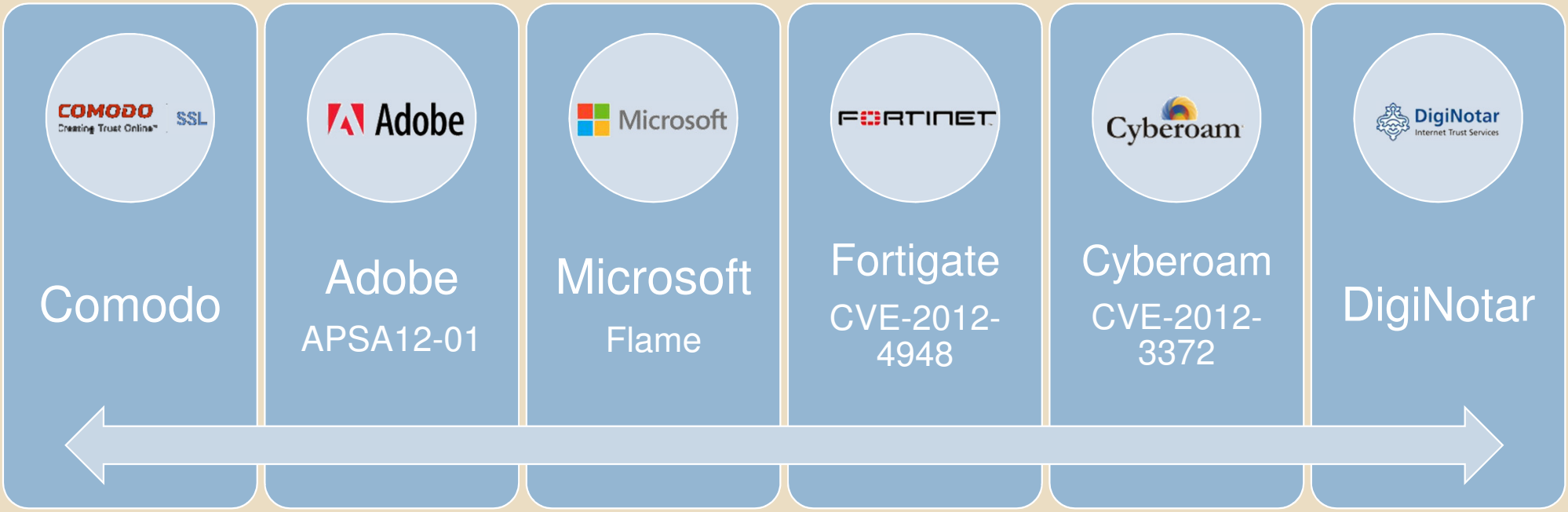
Predicative Analysis

- AV, Malwarebytes→ Open a Ticket
- Content Analysis- Keywords,

ADDRESSING SSL/TLS RISKS
WITH BRO-IDS

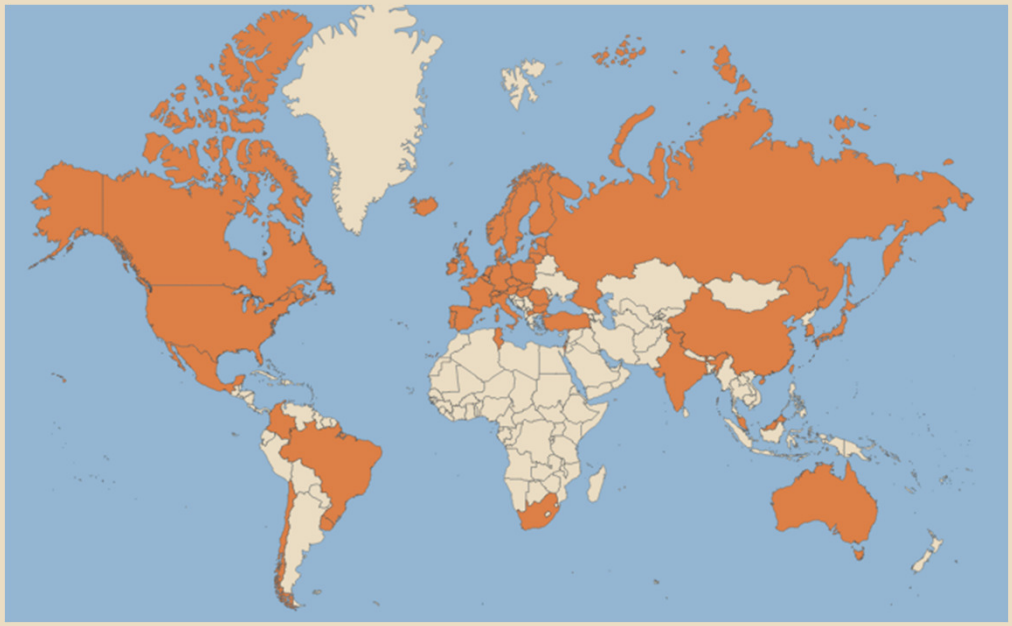
PUBLIC SSL/TLS EXPLOITS

A failing web of trust...



JURISDICTIONAL RISK

Distribution



Certificate Authority Entities

- 651 CA Organizations
- 52 Jurisdictions (Countries)
  - Many other Sub-CA

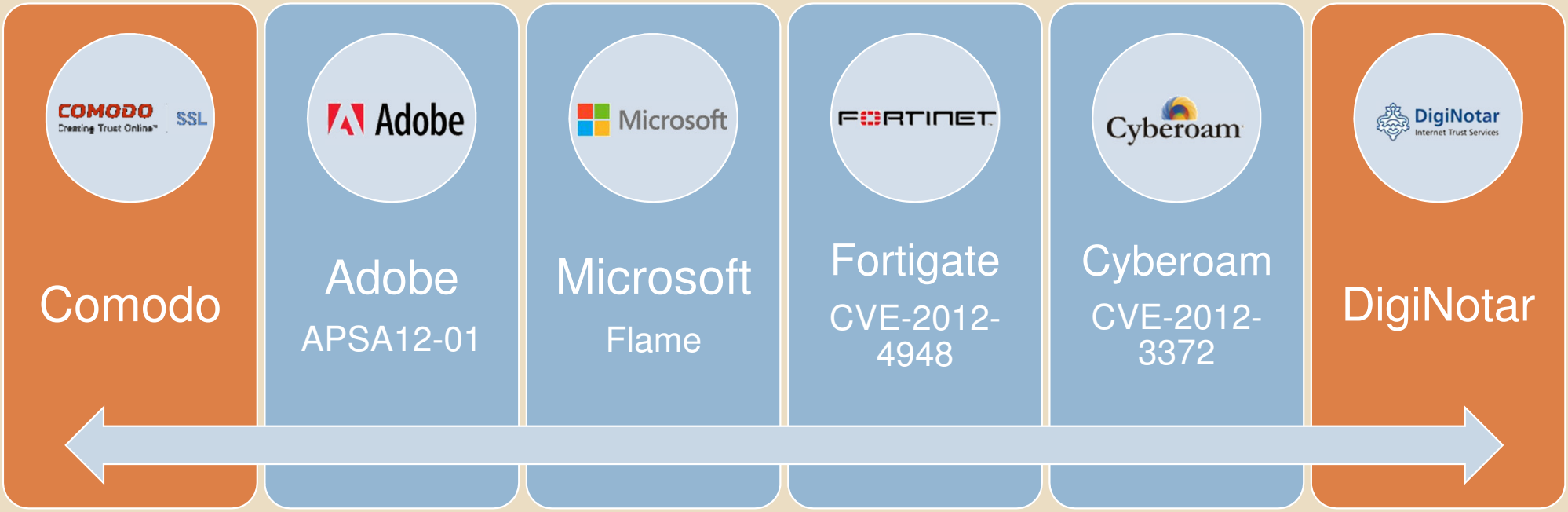
['AE', 'AT', 'AU', 'BE', 'BG', 'BM', 'BR', 'CA', 'CH', 'CL', 'CN', 'CO', 'CZ', 'DE', 'DK', 'EE', 'ES', 'EU', 'FI', 'FR', 'GB', 'HK', 'HU', 'IE', 'IL', 'IN', 'IS', 'IT', 'JP', 'KR', 'LT', 'LV', 'MK', 'MO','MX', 'MY', 'NL', 'NO', 'PL', 'PT', 'RO', 'RU', 'SE', 'SG', 'SI', 'SK', 'TN', 'TR', 'TW', 'UK', 'US', 'UY', 'WW', 'ZA']

Compiled by EFF SSL Observatory

PUBLIC SSL/TLS EXPLOITS

All 651 CA's can sign *everywhere* for *anything*.

The compromised companies are not the final target.



NIST WARNING

iTL

ITL BULLETIN FOR JULY 2012

Preparing for and Responding to Certification Authority Compromise and Fraudulent Certificate Issuance

Paul Turner, Venafi

William Polik, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce

Elaine Barker, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce

1. Executive Summary

As the use of Public Key Infrastructure (PKI) and digital certificates (e.g., the use of Transport Layer Security [TLS] and Secure Sockets Layer [SSL]) for the security of systems has increased, the certification authorities (CAs) that issue certificates have increasingly become targets for sophisticated cyber-attacks. In 2011, several public certification authorities were attacked, and at least two attacks resulted in the successful issuance of fraudulent certificates by the attackers. An attacker who breaches a CA to generate and obtain fraudulent certificates does so to launch further attacks against other organizations or individuals. An attacker can also use fraudulent certificates to authenticate as another individual or system or to forge digital signatures.

These recent attacks on CAs make it imperative that organizations ensure they are using secure CAs and must also be prepared to respond to a CA compromise or issuance of a fraudulent certificate. Responding to a CA compromise may require replacing all user or device certificates or trust anchors.[1] If an organization is not prepared with an inventory of certificate locations and owners, the organization will not be able to respond in a timely manner and may experience significant interruption in its operations for an extended period of time. This document provides an overview of CA compromise and fraudulent certificate issuance scenarios and recommends steps for preparing for and responding to these incidents.

Many organizations have certificates issued from an external CA, and some organizations operate their own CAs. Nearly all organizations have users and/or systems that establish security using certificates belonging to the parties with whom they communicate. Since many of today's applications are sold with installed trust anchors that users may not be aware of or

[1] Relying parties use root certificates, referred to as trust anchors in this document, that they store locally to verify certificates they receive.

1

NIST National Institute of Standards and Technology / U.S. Department of Commerce

CA Compromises

"An attacker who breaches a CA to generate and obtain fraudulent certificates does so

to launch further attacks against other organizations or individuals."

http://csrc.nist.gov/publications/nistbul/july-2012_itl-bulletin.pdf

A TALE OF TWO CERT(IES)

When both valid, which CERT to Trust?

```
-----BEGIN CERTIFICATE-----
MIIDgDCCAumgAwIBAgIKGI35CwAAAAB4CzANBgkqhkiG9w0BAQUFADBGMQswCQYD
VQQGEwJVUzETMBEGA1UEChMKR29vZ2xlIEluYzEiMCAGA1UEAxMZR29vZ2xlIElu
dGVybmV0IEF1dGhvcml0eTAeFw0xMzAxMDMxMjE1NTJaFw0xMzA2MDcxOTQzMjda
MGgxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRYwFAYDVQQHEw1N
b3VudGFpbiBWaWV3MRMwEQYDVQQKEwpHb29nbGUgSW5jMRcwFQYDVQQDEw53d3cu
Z29vZ2xlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAp0uFsoDllANv
ykrlbKlxgKFn97lG6Ca16b1ZT3vdGlBoxzrfcxXOqGkA1CcJqc3h0W4txqPpO9aq
lGODGmQnv/6HkNTmuOSJqHYjFRPgJ2s4CvofsexxCuw0/w2cHKfWRw/scGwqa4mQ
9d5Y6U6uTW/w8cp9csB6eZQo/oUBWMkCAwEAAaOCAVEwggFNMB0GA1UdJQQWMBQG
CCsGAQUFBwMBBggrBgEFBQcDAjAdBgNVHQ4EFgQUnkW9Yw+kcEJIu1VoSIQ8dwfb
6JQwHwYDVR0jBBgwFoAUv8Aw6/VDET5nup6R+/xq2uNrEiQwWwYDVR0fBFQwUjBQ
oE6gTIZKaHR0cDovL3d3dy5nc3RhdGljLmNvbS9Hb29nbGVJbnRlcm5ldEF1dGhv
cml0eS9Hb29nbGVJbnRlcm5ldEF1dGhvcml0eS5jcmwwZgYIKwYBBQUHAQEEWjBY
MFYGCCsGAQUFBzAChkpodHRwOi8vd3d3LmdzdGF0aWMuY29tL0dvb2dsZUludGVy
bmV0QXV0aG9yaXR5L0dvb2dsZUludGVybmV0QXV0aG9yaXR5LmNydDAMBgNVHRMB
Af8EAjAAMBkGA1UdEQQSMBCCDnd3dy5nb29nbGUuY29tMA0GCSqGSIb3DQEBBQUA
A4GBAFjwEoRMraJ+bM81lTrnT/qXXV1A2JwE+slBdVUysd4xAeg+yKnpxvfZ2H/i
AxELBVfQLO5R4f+Vr6axNFv4c8ne+FT4ZyNCEyD0sspESwhZXuXupc4ZMzm9xFa0
lxea+NUbP1EEgjiXkbtV6hcFVjFVgx7LsnSbuzp/SS418OFl
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
MIIFKDCCBBCgAwIBAgIQBeLmpM0J6lTWZbB1/iKiVjANBgkqhkiG9w0BAQUFADBm
MQswCQYDVQQGEwJOTDESMBAGA1UEChMJRGlnaU5vdGFyMSEwHwYDVQQDExhEaWdp
Tm90YXIgUHVibGljIENBIDIwMjUxIDAeBgkqhkiG9w0BCQEWEWluZm9AZGlnaW5v
dGFyLm5sMB4XDTExMDcxMDE5MDYzMFoXDTEzMDcwOTE5MDYzMFowajELMAkGA1UE
BhMCVVMxEzARBgNVBAoTCkdvb2dsZSBJbmMxFjAUBgNVBAcTDU1vdW50YWluIFZp
ZXcxFzAVBgNVBAUTDlBLMDAwMjI5MjAwMDAyMRUwEwYDVQQDEwwqLmdvb2dsZS5j
b20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDNbeKubCV0aCxhOiOS
CSQ/w9HXTYuD5BLKuiqXNw3setdTymeJz2L8aWOHo3nicFNDVwWTgwWomGNr2J6Q
7gliINNSW0rR4E1l2szRkcnAY6c6i/Eke93nF4i2hDsnIBveolF5yjpuRm73uQQD
ulHjA3BFRF/PTi0fw2/Yt+8ieoMuNcMWN6Eou5Gqt5YZkWv176ofeCbsBmMrP87x
OhhtTDckCapk4VQZG2XrfzZcV6tdzCp5TI8uHdul7cdzXmlimZ8tyvzFeiCEOQN8
vPNzB/fIr3CJQ5q4uM5aKT3DD5PeVzf4rfJKQNgCTWiIBc9XcWEUuszwAsnmg7e2
EJRdAgMBAAGjggHMMIIByDA6BggrBgEFBQcBAQQuMCwwKgYIKwYBBQUHMAGGHmh0
dHA6Ly92YWxpZGF0aW9uLmRpZ2lub3Rhci5ubDAfBgNVHSMEGDAWgBTfM8Cvkv43
/LbYFhbQ2bGR1fpupTAJBgNVHRMEAjAAMIHGBgNVHSAEgb4wgbswgbgGDmCEEAGH
aQEBAQIEAQICMIG1MCcGCCsGAQUFBwIBFhtodHRwOi8vd3d3LmRpZ2lub3Rhci5u
bC9jcHMwegYIKwYBBQUHAgIwbhpsQ29uZGl0aW9ucywgYXMgbWVudGlvbmVkIG9u
IG91ciB3ZWJzaXRlICh3d3cuZGlnaW5vdGFyLm5sKSwgYXJlIGFwcGxpY2FibGUg
dG8gYWxsIG91ciBwcm9kdWN0cyBhbmQgc2VydmljZXMuMEkGA1UdHwRCMEAwPqA8
oDqGOGh0dHA6Ly9zZXJ2aWNlLmRpZ2lub3Rhci5ubC9jcmwvcHVibGljMjAyNS9s
YXRlc3RDUkwuY3JsMA4GA1UdDwEB/wQEAwIEsDAbBgNVHREEFDASgRBhZG1pbkBn
b29nbGUuY29tMB0GA1UdDgQWBBQHSn0WJzIo0eMBMQUNsMqN6eF/7TANBgkqhkiG
9w0BAQUFAAOCAQEAAs5dL7N9wzRJkI4Aq4lC5t8j5ZadqnqUcgYLADzSv4ExytNH
UY2nH6iVTihC0UPSsILWraoeApdT7Rphz/8DLQEBRGdeKWAptNM3EbiXtQaZT2uB
pidL8UoafX0kch3f71Y1scpBEjvu5ZZLnjg0A8AL0tnsereOVdDpU98bKqdbbrnM
FRmBlSf7xdaNca6JJHeEpga4E9Ty683CmccrSGXdU2tTCuHEJww+iOAUtPIZcsum
U7/eYeY1pMyGLyIjbNgRY7nDzRwvM/BsbL9eh4/mSQj/4nncqJd22sVQpCggQiVK
baB2sVGcVNBkK55bT8gPqnx8JypubyUvayzZGg==
-----END CERTIFICATE-----
```

A TALE OF TWO CERT(IES)

When both valid, which CERT to Trust?


```
-----BEGIN CERTIFICATE-----
MIIDgDCCAumgAwIBAgIKGI35CwAAAAB4CzANBgkqhkiG9w0BAQUFADBGMQswCQYD
VQQGEwJVUzETMBEGA1UEChMKR29vZ2xlIEluYzEiMCAGA1UEAxMZR29vZ2xlIElu
dGVybmV0IEF1dGhvcml0eTAeFw0xMzAxMDMxMjE1NTJaFw0xMzA2MDcxOTQzMjda
MGgxCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMRYwFAYDVQQHEw1N
b3VudGFpbiBWaWV3MRMwEQYDVQQKEwpHb29nbGUgSW5jMRcwFQYDVQQDEw53d3cu
Z29vZ2xlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAp0uFsoDllANv
ykrlbKlxgKFn97lG6Ca16b1ZT3vdGlBoxzrfcxXOqGkA1CcJqc3h0W4txqPpO9aq
lGODGmQnv/6HkNTmuOSJqHYjFRPgJ2s4CvofsexxCuw0/w2cHKfWRw/scGwqa4mQ
9d5Y6U6uTW/w8cp9csB6eZQo/oUBWMkCAwEAAaOCAVEwggFNMB0GA1UdJQQWMBQG
CCsGAQUFBwMBBggrBgEFBQcDAjAdBgNVHQ4EFgQUnkW9Yw+kcEJIu1VoSIQ8dwfb
6JQwHwYDVR0jBBgwFoAUv8Aw6/VDET5nup6R+/xq2uNrEiQwWwYDVR0fBFQwUjBQ
oE6gTIZKaHR0cDovL3d3dy5nc3RhdGljLmNvbS9Hb29nbGVJbnRlcm5ldEF1dGhv
cml0eS9Hb29nbGVJbnRlcm5ldEF1dGhvcml0eS5jcmwwZgYIKwYBBQUHAQEEWjBY
MFYGCCsGAQUFBzAChkpodHRwOi8vd3d3LmdzdGF0aWMuY29tL0dvb2dsZUludGVy
bmV0QXV0aG9yaXR5L0dvb2dsZUludGVybmV0QXV0aG9yaXR5LmNydDAMBgNVHRMB
Af8EAjAAMBkGA1UdEQQSMBCCDnd3dy5nb29nbGUuY29tMA0GCSqGSIb3DQEBBQUA
A4GBAFjwEoRMraJ+bM81lTrnT/qXXV1A2JwE+slBdVUysd4xAeg+yKnpxvfZ2H/i
AxELBVfQLO5R4f+Vr6axNFv4c8ne+FT4ZyNCEyD0sspESwhZXuXupc4ZMzm9xFa0
lxea+NUbPlEEgjiXkbtV6hcFVjFVgx7LsnSbuzp/SS418OFl
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
MIIFKDCCBBCgAwIBAgIQBeLmpM0J6lTWZbB1/iKiVjANBgkqhkiG9w0BAQUFADBm
MQswCQYDVQQGEwJOTDESMBAGA1UEChMJRGlnaU5vdGFyMSEwHwYDVQQDExhEaWdp
Tm90YXIgUHVibGljIENBIDIwMjUxIDAeBgkqhkiG9w0BCQEWEWluZm9AZGlnaW5v
dGFyLm5sMB4XDTExMDcxMDE5MDYzMFoXDTEzMDcwOTE5MDYzMFowajELMAkGA1UE
BhMCVVMxEzARBgNVBAoTCkdvb2dsZSBJbmMxFjAUBgNVBAcTDU1vdW50YWluIFZp
ZXcxFzAVBgNVBAUTDlBLMDAwMjI5MjAwMDAyMRUwEwYDVQQDEwwqLmdvb2dsZS5j
b20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDNbeKubCV0aCxhOiOS
CSQ/w9HXTYuD5BLKuiqXNw3setdTymeJz2L8aWOHo3nicFNDVwWTgwWomGNr2J6Q
7gliINNSW0rR4E1l2szRkcnAY6c6i/Eke93nF4i2hDsnIBveolF5yjpuRm73uQQD
ulHjA3BFRF/PTi0fw2/Yt+8ieoMuNcMWN6Eou5Gqt5YZkWv176ofeCbsBmMrP87x
OhhtTDckCapk4VQZG2XrfzZcV6tdzCp5TI8uHdul7cdzXmlimZ8tyvzFeiCEOQN8
vPNzB/fIr3CJQ5q4uM5aKT3DD5PeVzf4rfJKQNgCTWiIBc9XcWEUuszwAsnmg7e2
EJRdAgMBAAGjggHMMIIByDA6BggrBgEFBQcBAQQuMCwwKgYIKwYBBQUHMAGGHmh0
dHA6Ly92YWxpZGF0aW9uLmRpZ2lub3Rhci5ubDAfBgNVHSMEGDAWgBTfM8Cvkv43
/LbYFhbQ2bGR1fpupTAJBgNVHRMEAjAAMIHGBgNVHSAEgb4wgbswgbgGDmCEEAGH
aQEBAQIEAQICMIG1MCcGCCsGAQUFBwIBFhtodHRwOi8vd3d3LmRpZ2lub3Rhci5u
bC9jcHMwegYIKwYBBQUHAgIwbhpsQ29uZGl0aW9ucywgYXMgbWVudGlvbmVkIG9u
IG91ciB3ZWJzaXRlICh3d3cuZGlnaW5vdGFyLm5sKSwgYXJlIGFwcGxpY2FibGUg
dG8gYWxsIG91ciBwcm9kdWN0cyBhbmQgc2VydmljZXMuMEkGA1UdHwRCMEAwPqA8
oDqGOGh0dHA6Ly9zZXJ2aWNlLmRpZ2lub3Rhci5ubC9jcmwvcHVibGljMjAyNS9s
YXRlc3RDUkwuY3JsMA4GA1UdDwEB/wQEAwIEsDAbBgNVHREEFDASgRBhZG1pbkBn
b29nbGUuY29tMB0GA1UdDgQWBBQHSn0WJzIo0eMBMQUNsMqN6eF/7TANBgkqhkiG
9w0BAQUFAAOCAQEAAs5dL7N9wzRJkI4Aq4lC5t8j5ZadqnqUcgYLADzSv4ExytNH
UY2nH6iVTihC0UPSsILWraoeApdT7Rphz/8DLQEBRGdeKWAptNM3EbiXtQaZT2uB
pidL8UoafX0kch3f71Y1scpBEjvu5ZZLnjg0A8AL0tnsereOVdDpU98bKqdbbrnM
FRmBlSf7xdaNca6JJHeEpga4E9Ty683CmccrSGXdU2tTCuHEJww+iOAUtPIZcsum
U7/eYeYlpMyGLyIjbNgRY7nDzRwvM/BsbL9eh4/mSQj/4nncqJd22sVQpCggQiVK
baB2sVGcVNBkK55bT8gPqnx8JypubyUvayzZGg==
-----END CERTIFICATE-----
```

WEAK HASH

Known Attacks

- Additional risk Enterprises should control & monitor
- Collision Attacks

Replace Immediately

- MD2
- MD4
- MD5

```
Risk factor :

Medium / CVSS Base Score : 4.0
(CVSS2#AV:N/AC:H/Au:N/C:P/I:P/A:N)
CVSS Temporal Score : 3.3
(CVSS2#E:F/RL:OF/RC:C)
Public Exploit Available : true
```

MITIGATIONS EFFORTS

Well known problem

CMU Perspectives

- Browser Based

Certificate Patrol

- Browser Based
- Notify on Updates

convergence.io

- Browser Based
- Distributed Trust

ISCI SSL Notary

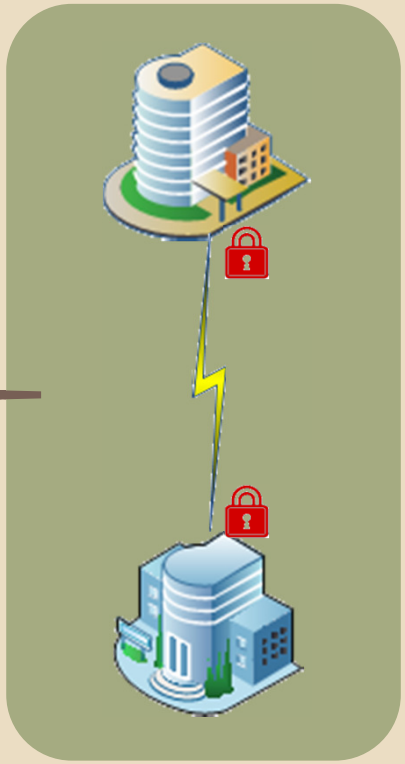
- DNS Lookups

AFFECTED SERVICES

Example Use Cases

+ Credit Checks
+ Authorization and Accounting
+ Supply Chain Management
+ e-Commerce
+ Marketing

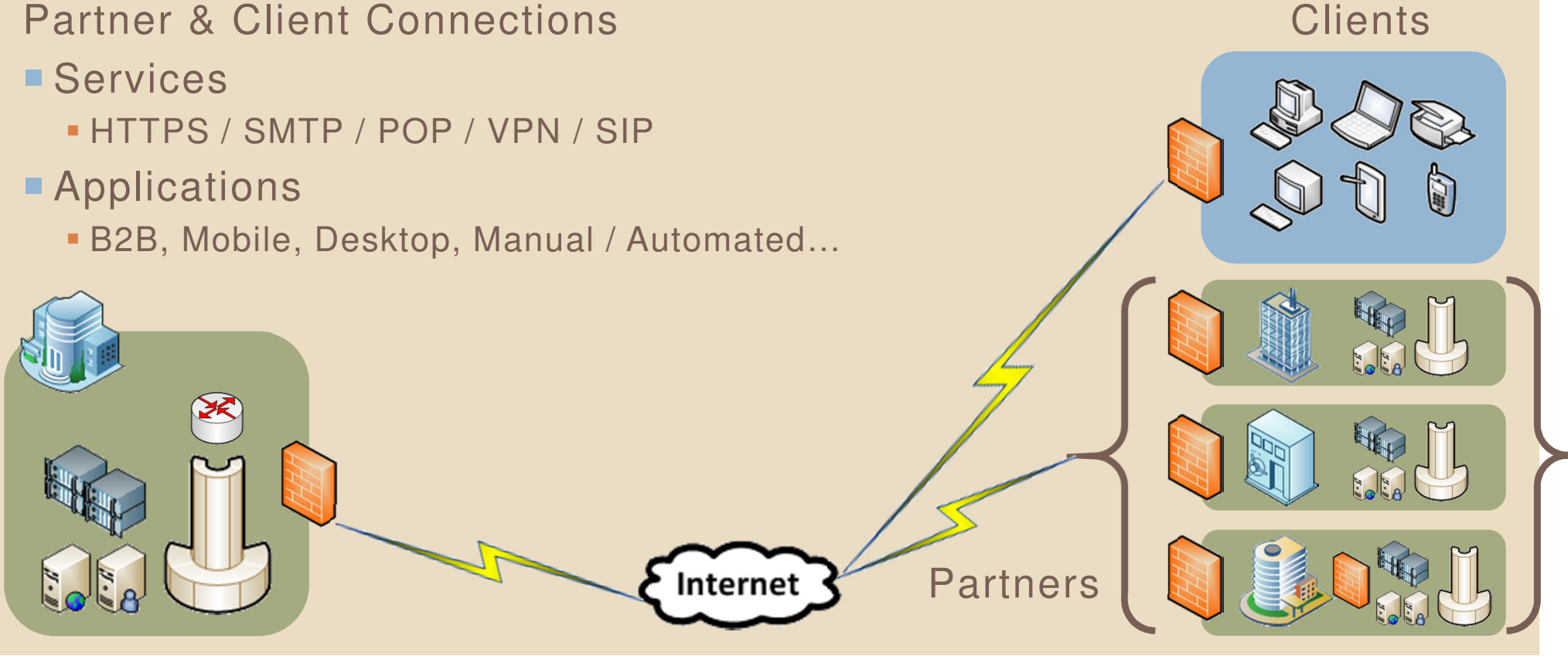
HTTPS	SMTP POP/IMAP	SSL/TLS VPN	SIP



B2B WHAT SHOULD WE KNOW

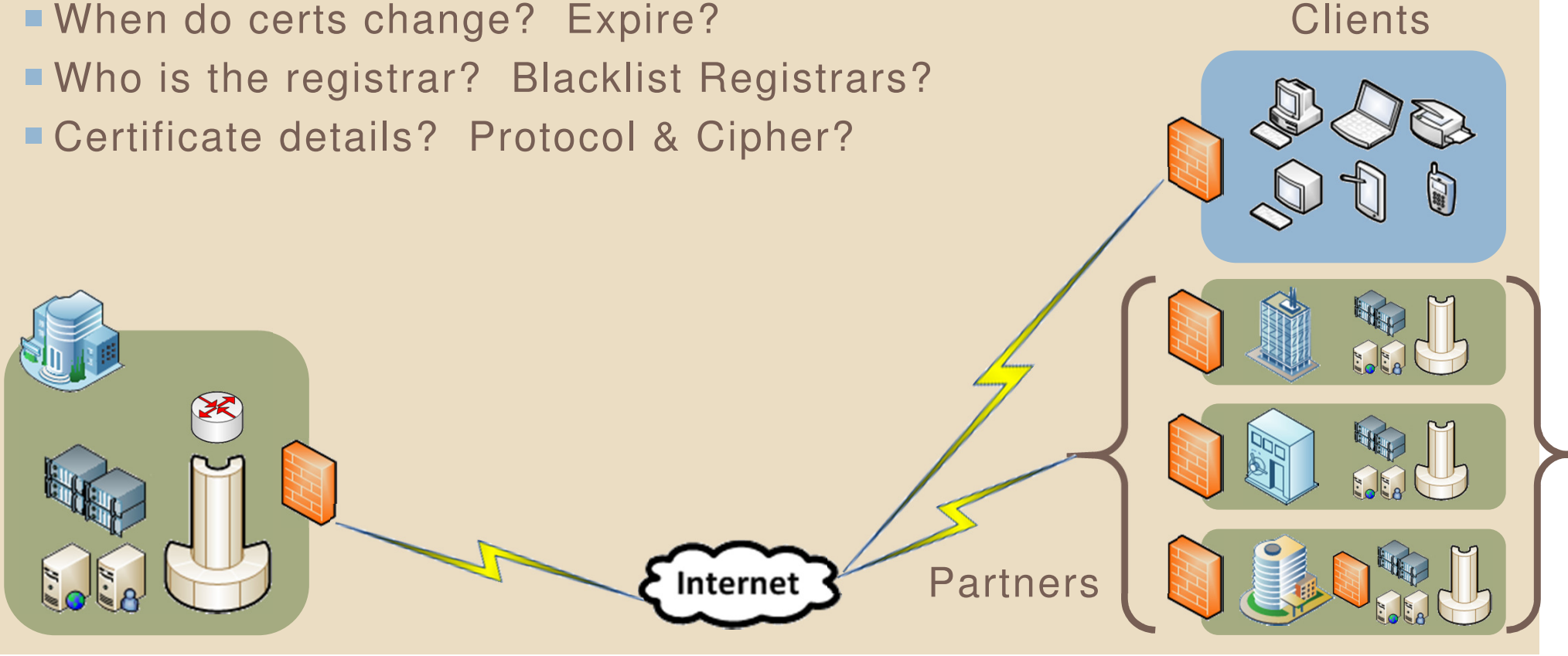
Partner & Client Connections

- Services
  - HTTPS / SMTP / POP / VPN / SIP
- Applications
  - B2B, Mobile, Desktop, Manual / Automated…



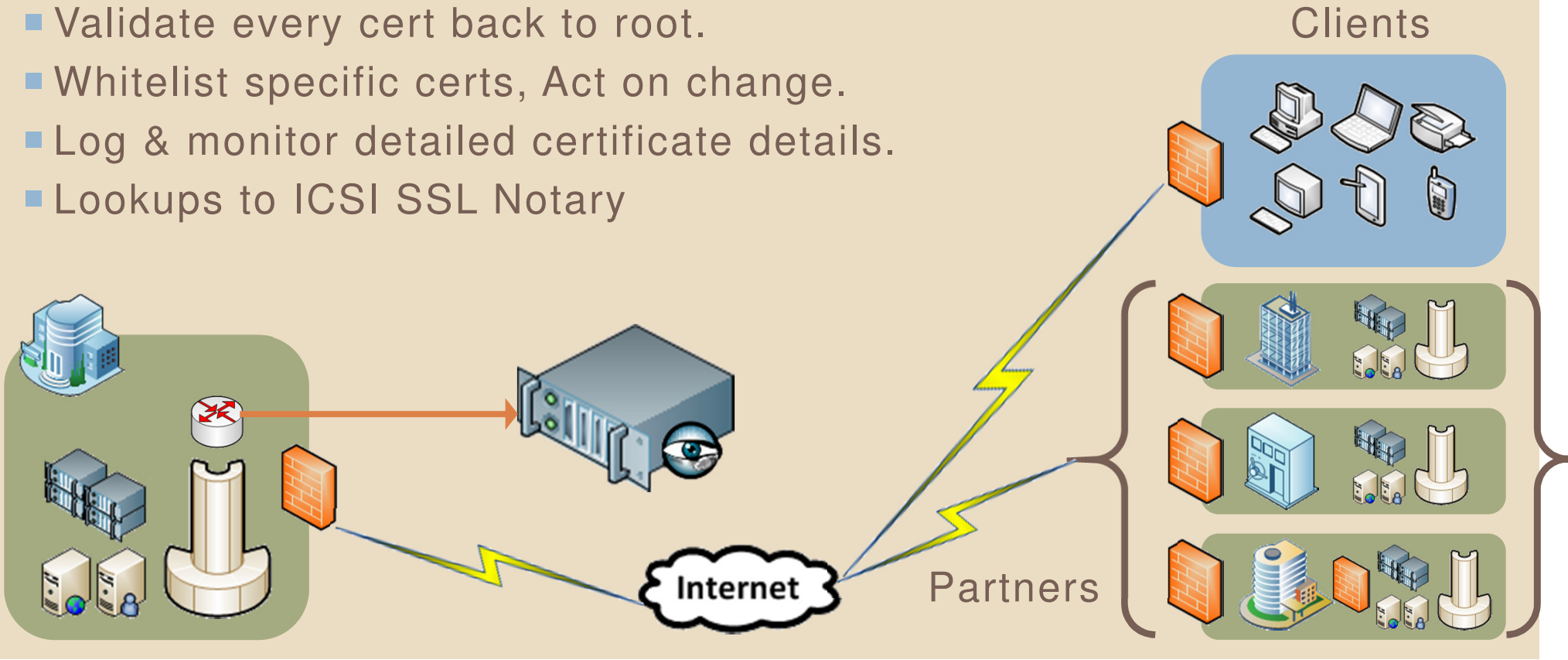
B2B SSL/TLS IOC

- When do certs change? Expire?
- Who is the registrar? Blacklist Registrars?
- Certificate details? Protocol & Cipher?



BRO-IDS INSIGHTS

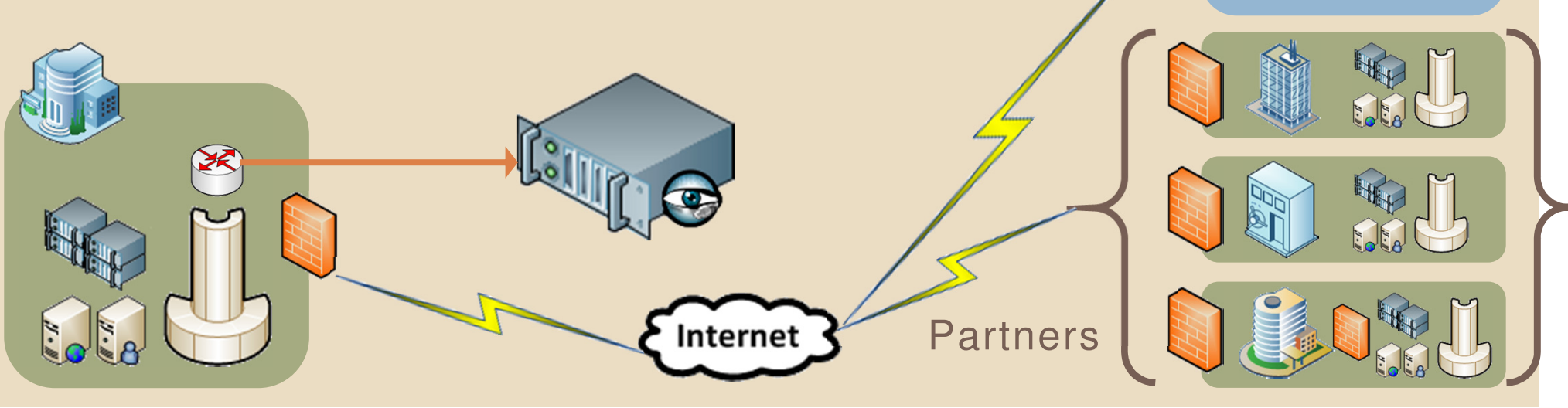
- Validate every cert back to root.
- Whitelist specific certs, Act on change.
- Log & monitor detailed certificate details.
- Lookups to ICSI SSL Notary



BRO-IDS INSIGHTS

< DEMONSTRATION >

- Bro-IDS, validating keys
- Bro-IDS, signing keys back to root
- Bro-IDS, whitelisting and alerting on keys



SSL ATTACKS

Lucky Thirteen: Breaking the TLS and DTLS Record Protocols

Nadhem J. AlFardan and Kenneth G. Paterson*
Information Security Group
Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK
{nadhem.alfardan.2009, kenny.paterson}@rhul.ac.uk

6th February 2013

Abstract

The Transport Layer Security (TLS) protocol aims to provide confidentiality and integrity of data in transit across untrusted networks. TLS has become the de facto *secure protocol of choice for Internet and mobile applications. DTLS is a variant of TLS that is growing in importance. In this paper, we present distinguishing and plaintext recovery attacks against TLS and DTLS. The attacks are based on a delicate timing analysis of decryption processing in the two protocols. We include experimental results demonstrating the feasibility of the attacks in realistic network environments for several different implementations of TLS and DTLS, including the leading OpenSSL implementations. We provide countermeasures for the attacks. Finally, we discuss the wider implications of our attacks for the cryptographic design used by TLS and DTLS.*

Keywords TLS, DTLS, CBC-mode encryption, timing attack, plaintext recovery

1 Introduction

TLS is arguably the most widely-used secure communications protocol on the Internet today. Starting life as SSL, the protocol was adopted by the IETF and specified as TLS 1.0 [10]. It has since evolved through TLS 1.1 [11] to the current version TLS 1.2 [12]. Various other RFCs define additional TLS cryptographic algorithms and extensions. TLS is now used for securing a wide variety of application-level traffic and has become a serious rival to IPsec for general VPN usage. It is widely supported in client and server software and in cryptographic libraries for embedded systems, mobile devices, and web application frameworks. Open-source implementations of TLS and DTLS include OpenSSL, GnuTLS, PolarSSL and CyaSSL.

The DTLS protocol is a close relative of TLS, developed from TLS by making minimal changes so as to allow it to operate over UDP instead of TCP [25]. This makes DTLS suitable for use where the costs of TCP connection establishment and TCP retransmissions are not warranted, for example, in voice and gaming applications. DTLS exists in two versions, DTLS 1.0 [31], which roughly matches TLS 1.1 and DTLS 1.2 [32], which aligns with TLS 1.2.

Both TLS and DTLS are actually protocol suites, rather than single protocols. The main component of (D)TLS that concerns us here is the Record Protocol, which uses symmetric key cryptography (block ciphers, stream ciphers and MAC algorithms) in combination with sequence numbers to build a secure channel for transporting application-layer data. Other major components are the (D)TLS Handshake Protocol, which is responsible for authentication, session key establishment and ciphersuite negotiation, and the TLS Alert Protocol, which carries error messages and management traffic. Setting aside dedicated authenticated encryption algorithms (which are yet to see widespread support in TLS or DTLS implementations), the (D)TLS Record Protocol uses a MAC-Encode-Encrypt (MEE) construction. Here, the plaintext data to be transported is first passed through a MAC algorithm (along with certain header bytes) to create a MAC tag. The supported MAC algorithms are all HMAC-based, with MD5, SHA-1 and SHA-256 being the allowed hash algorithms in TLS 1.2 [12]. Then an encoding step takes place. For the RC4 stream cipher, this just involves concatenation of the plaintext and the MAC tag, while for CBC-mode encryption (the other possible option), the plaintext, MAC tag, and some encryption padding of a specified format are concatenated. In the encryption step, the encoded plaintext is encrypted with the selected cipher. In the case where CBC-mode is selected, the block cipher is DES, 3DES or AES (with DES being deprecated in TLS 1.2). Following [28], we refer to this MEE construction as MEE-TLS-CBC. We provide greater detail on its operation in the (D)TLS Record Protocol in Section 2.

The widespread use of TLS (and the increasing use of DTLS) makes the continued study of the security of these protocols of great importance. Indeed, the evolution of the TLS Record Protocol has largely been driven by cryptographic attacks that have been discovered against it, including those in [37, 6, 26, 2, 3, 13, 28, 1].

Of particular interest lately have been attacks based on the use of chained initialisation vectors (IVs) for CBC-mode in SSL and TLS 1.0, in particular, the so-called BEAST attack [13] which has its roots in [35, 26, 2, 3]. This attack achieved full plaintext recovery against TLS, but only in scenarios where an attacker can gain access to a chosen plaintext capability, per-

*This author's research supported by an EPSRC Leadership Fellowship, EP/H005455/1.

Crypto is Hard

- 2011 BEAST
  - Chained IVs in CBC-mode in SSL/TLS 1.0
- 2012 CRIME
  - Compression
- 2013 LUCKY 13
  - Timing Attack
  - Wide Vulnerability
  - TLS 1.0 / 1.1 / 1.2, DTLS 1.0 / 1.2, SSL 3.0

NEEDLE IN A HAYSTACK?

1

ssl_client_hello_count:	11
ssl_server_hello_count:	11
ssl_extension_count:	142
ssl_established_count:	11
ssl_alert_count:	0
ssl_ticket_handshake_count:	7
x509_certificate_count:	14
x509_extension_count:	0
x509_error_count:	0

2

ssl_client_hello_count:	12
ssl_server_hello_count:	12
ssl_extension_count:	128
ssl_established_count:	12
ssl_alert_count:	0
ssl_ticket_handshake_count:	6
x509_certificate_count:	21
x509_extension_count:	0
x509_error_count:	0

3

ssl_client_hello_count:	2
ssl_server_hello_count:	2
ssl_extension_count:	0
ssl_established_count:	2
ssl_alert_count:	0
ssl_ticket_handshake_count:	0
x509_certificate_count:	1
x509_extension_count:	0
x509_error_count:	0

4

ssl_client_hello_count:	4096
ssl_server_hello_count:	0
ssl_extension_count:	12288
ssl_established_count:	0
ssl_alert_count:	4
ssl_ticket_handshake_count:	0
x509_certificate_count:	0
x509_extension_count:	0
x509_error_count:	0

BRO SCRIPT
HTTP BRUTE FORCING

BRO SCRIPT APPROACH

Overview

- Current Attacks are Ridiculous
- Sum/Avg Protocol Metrics

"*Red teams aren't any better because they don't have to be.*"

Basic Steps

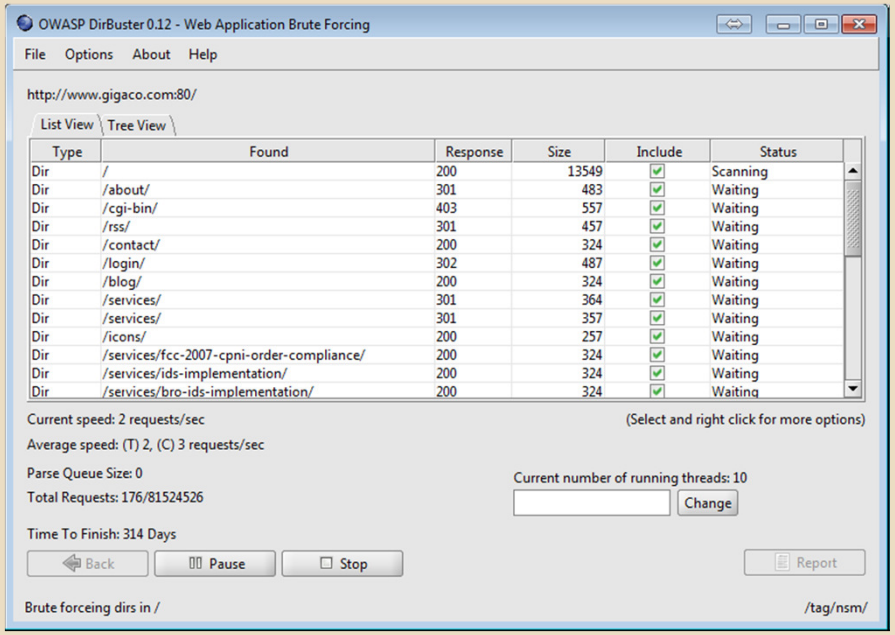
- Review Attack
- Hypothesis
- Algorithm

HTTP BRUTE FORCE

Overview

- Fuzz a website
- Discover Unknown Apps
- Response Codes
  - High Rate of 404's
- Look at an attack
  - Base Case & Extended

Attack



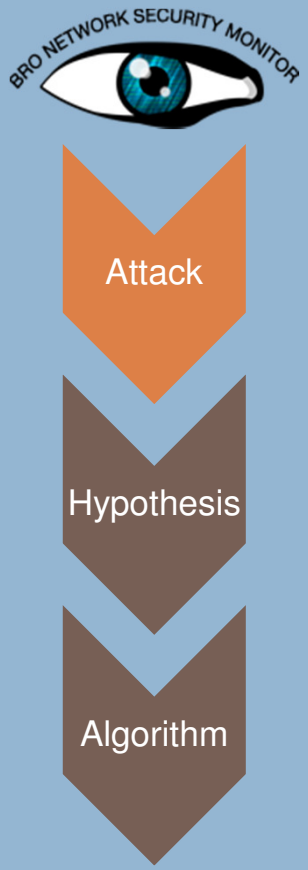
Client Side

- High Rate of Requests
- High Rate of 404
- Application Layer Semantic Analysis

- Distributed Scans?
- Slow Scans?

Server Side

- High Rate of 404
  - Errors <> attack
- Code <> End



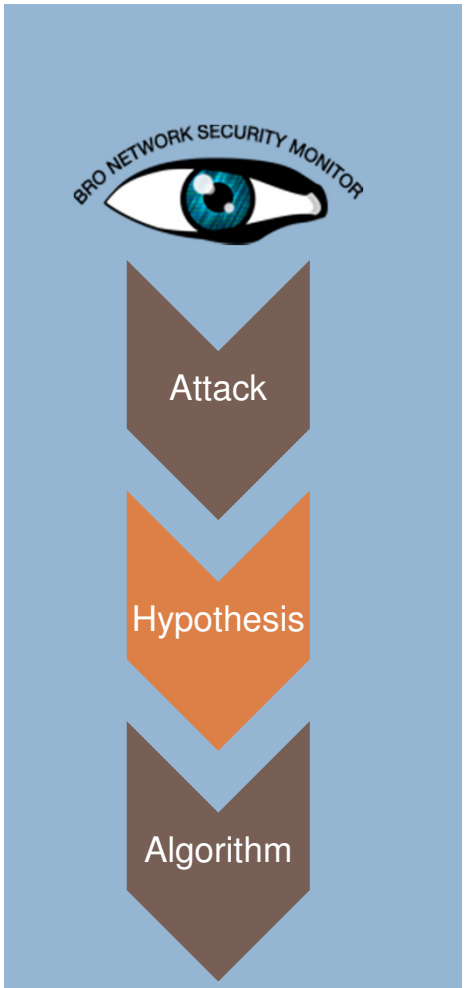
We could track valid URI's

Could we tack invalid URI's?

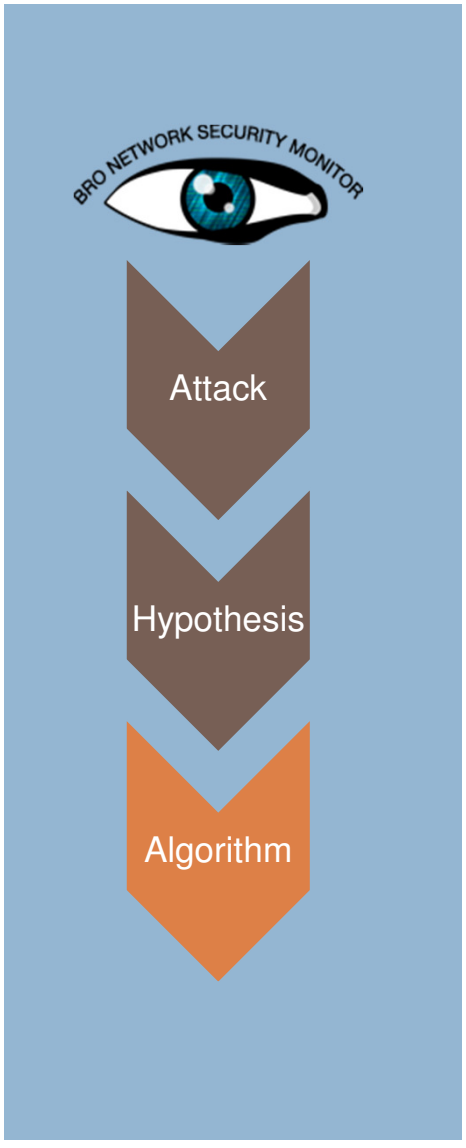
Could we track rate of requests?

Could we track http status codes?

- Scaleable? To 10 Gig?
- What conditions will it detect?
- Metrics framework?

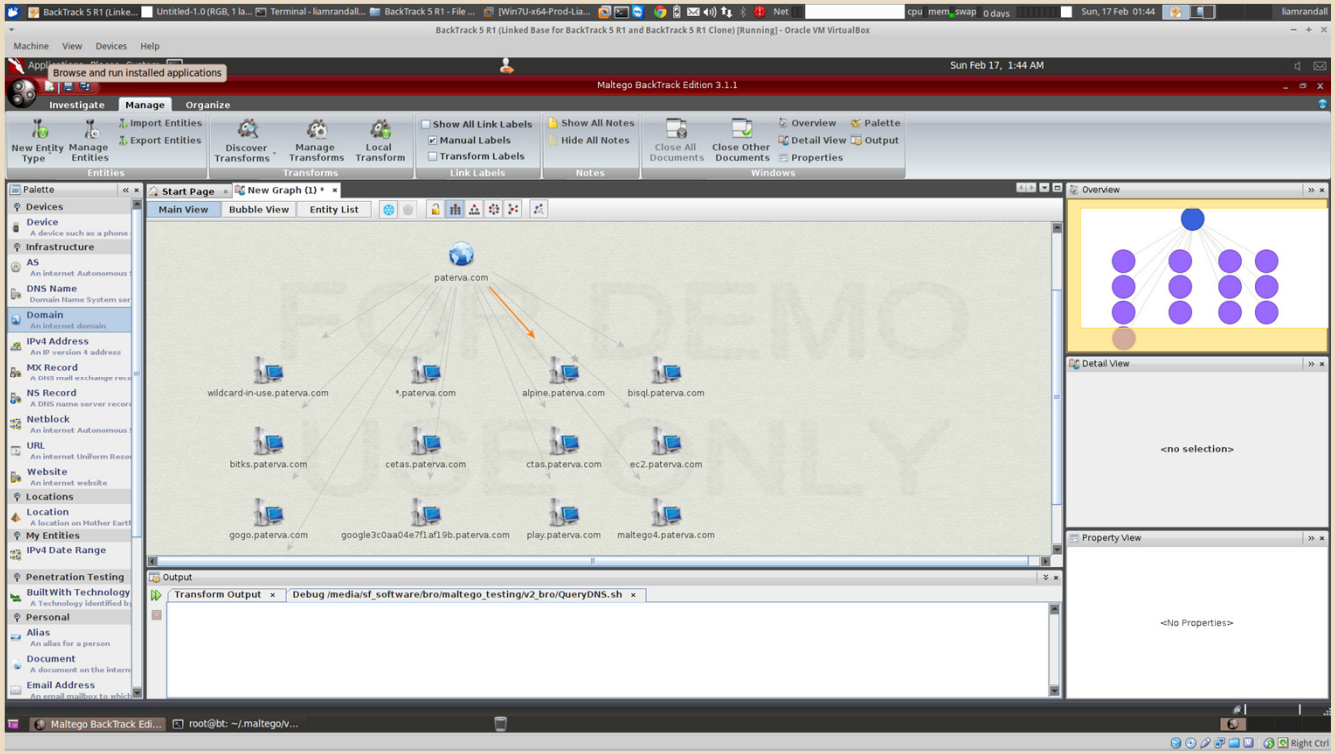


```
StatusCodeWhitelist table[count]

# table of servers? sites?
# table of clients?
    # by site
        # by status code
            # count

< coding demonstration >
```



BROTEGO
MALTEGO & BRO-IDS

BROTEGO



- Historical Analysis
- DGA/Fast Flux
- Atribution
- +cool
- -slow at scale
- - Parallels, Elastic Search Client, etc.

SPECIAL THANKS

- Katie Randall (patient and loving wife)

- Bro Team
  - Seth Hall (ICSI)
  - Robin Sommer (ICSI)
  - Vern Paxson (UC Berkeley)
- Shmoocon
  - Bruce and Heidi Potter
  - Shmoolabs & staff
- Friends & Colleages
  - DuplictyCTF Crew, #snort-gui, #derbycon