

Change Management:

DYNAMIC NETWORK MAPPING

LinuxWorld San Francisco | Security Track

Presented by Joshua D. Abraham

August 16th 2006

jabra@ccs.neu.edu | Northeastern University

Agenda



- How do we scan?
- What are the limitations of the tools?
 - Nmap and Xprobe
- What are new ways to handle these limitations?
 - PBNJ 1.0
 - PBNJ 2.0

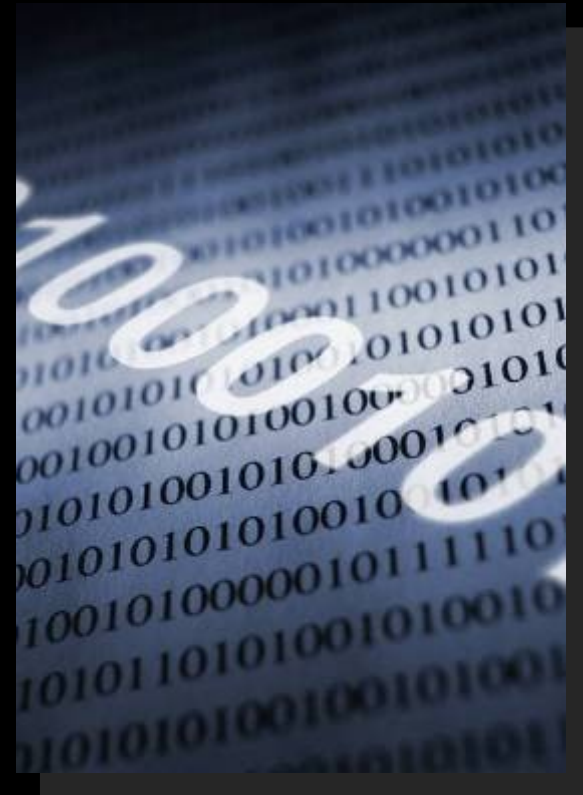
Knowledge is Everything



- What your machines are running?
- What other machines on your network are running?
- If you don't know your network & when it's changing, you're not secure
- Changes that occur ex:
 - Rogue FTP service
 - Web server keeps crashing

Who Needs to Know this?

- Network Managers
- Unix Admins
- Windows Admins
- Network Admins
- Security Professionals



Current Technologies

- Active Scanners
 - Network Mapping
 - (Nmap & Xprobe)
 - Vulnerability Scanner
 - (Nikto & Nessus)
 - Application Mapping
 - (Nmap & Amap)
- Passive Scanners
 - P0f, PADS, etc



Active Scanners



- Scan only the targets you want
 - single target or range of targets
- Control over the scan
- XML Output (Nmap and Nessus)



- Probes
 - TCP SYN, TCP Connect, Xmas Tree, ACK
 - ICMP
 - echo (ping), timestamp req, info req or UDP closed port
- Compare the properties for each OS
 - database or matrix

Version Detection

```
$ telnet 127.0.0.1 22
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
SSH-2.0-OpenSSH_3.9p1 Debian-1ubuntu2.2
```

- Connection to port
 - soft banner or hard banner
- Probe service for banner (NULL, Get, help, etc)
- Compare banner and service list for a match using Regular expressions
- Then pull out the Version from the banner

Passive Scanners

- Can handle range of IPs and is not limited to specific target or targets
- Version Detection
 - banner grabbing
- Fingerprint



Limitations of Active Scanners

- Out of date instantly
- Loud – This can alert targets of you (IDS & logs)
- Quality of scan can be affected
 - firewalls, routers and targets firewalls
- Affect targets
 - TCP stack
 - state tables
 - logs

Nmap's Limitations

- Only a snapshot in time
- Banner grabbing isn't displayed
- Fingerprinting isn't very accurate
 - 4.20 >= are a lot better than previous versions
 - Still not perfect



Nmap Changes Demo (tkdiff)

test1.out vs. test2.out - TkDiff 4.0.2

File Edit View Mark Merge Help

1 : 2c2 Merge: Diff: Mark:

test1.out test2.out

```
1 | !Starting Nmap 4.20 ( http://insecure.org ) at 2007-01-16 1
2 | Interesting ports on localhost (127.0.0.1):
3 | !Not shown: 1695 closed ports
4 | PORT      STATE SERVICE VERSION
5 | 22/tcp    open  ssh      OpenSSH 4.2p1 Debian 7ubuntu3.1 (pro
6 |
7 | 631/tcp   open  ipp      CUPS 1.2
8 | Device type: general purpose
9 | Running: Linux 2.6.X
10 | OS details: Linux 2.6.14 - 2.6.16
11 | Uptime: 0.985 days (since Mon Jan 15 18:49:22 2007)
12 | Network Distance: 0 hops
13 | !Service Info: OS: Linux
14 |
15 | OS and Service detection performed. Please report any incor
16 | !Nmap finished: 1 IP address (1 host up) scanned in 8.390 s
```

```
1 | !Starting Nmap 4.20 ( http://insecure.org ) at 2007-01-16 1
2 | Interesting ports on localhost (127.0.0.1):
3 | !Not shown: 1694 closed ports
4 | PORT      STATE SERVICE VERSION
5 | 22/tcp    open  ssh      OpenSSH 4.2p1 Debian 7ubuntu3.1 (pro
6 |
7 | +25/tcp   open  smtp     Postfix smtpd
8 | 631/tcp   open  ipp      CUPS 1.2
9 | Device type: general purpose
10 | Running: Linux 2.6.X
11 | OS details: Linux 2.6.14 - 2.6.16
12 | Uptime: 0.985 days (since Mon Jan 15 18:49:22 2007)
13 | Network Distance: 0 hops
14 | !Service Info: Host: oreo; OS: Linux
15 |
16 | OS and Service detection performed. Please report any incor
17 | !Nmap finished: 1 IP address (1 host up) scanned in 8.314 s
```

1 of 5

Changes with Nmap

- tkdiff is ok for the people who like gui's
 - people who have the time for dealing with comparing files
- what about diff & grep ?

Nmap Changes by Hand 1

```
$ diff -u test1.out test2.out | grep -Eo \  
  "( (^\+[0-9].*) || (^-[0-9].*) )" \  
+25/tcp  open  smtp  Postfix smtpd
```

“Not bad but I want to be notified via email”

Nmap Changes by Hand 2

```
$ diff -u test1.out test2.out | grep -Eo \  
"((^\+[0-9].*)||(^-[0-9].*))" | \  
mail "Changes `date`" root
```


Nmap Changes by Hand 3

- Store the previous and the current in logical files

```
diff -u prev.out current.out | grep -Eo \  
"((^\+[0-9].*) || (^\-[0-9].*))" | \  
mail "Changes `date`" root
```

Issues Good and Bad

- Good
 - Works okay for a single IP
 - Email is a plus
 - Could be automated (requires scripting)
- Bad
 - Tedious
 - Does not work with IP ranges
 - Error prone
 - Not flexible

Xprobe's Limitations

- Lack of intelligence in scanning
- Database is out of date
- Based on ICMP probes
 - Can be and is often intentionally blocked
- Doesn't have the dev community Nmap does



Limitations of Passive Scanners

- Need privileges to sniff
- Encrypted or Tunnelled traffic hidden
- Identification mostly based on banner
- Does not work on a switched network, spanning ports are needed



“What if we store the information, so we can monitor changes over time?”

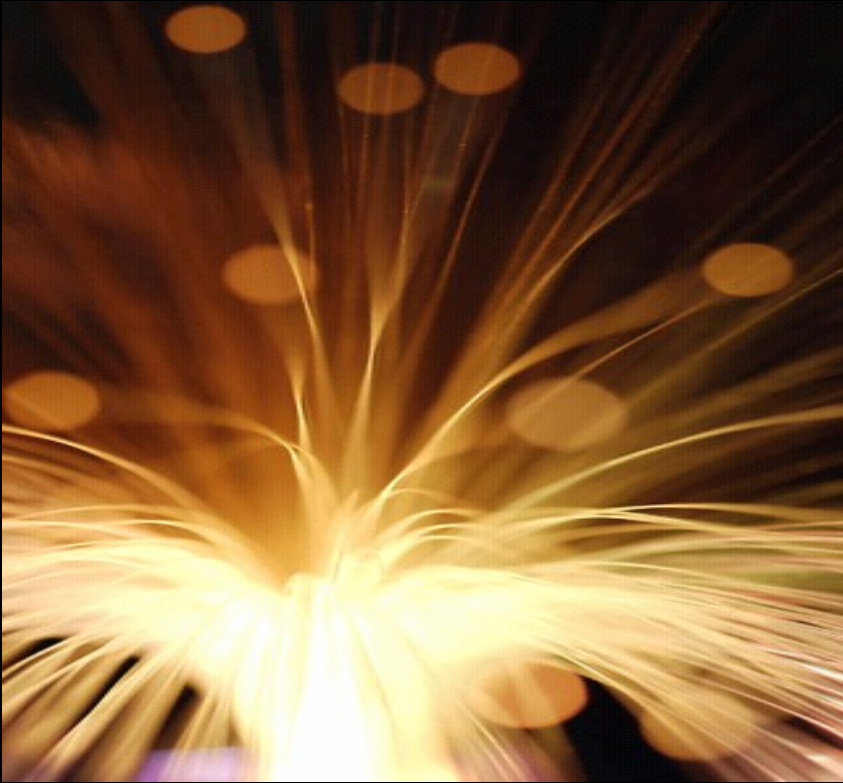
PBNJ 1.0

- First tool to monitors changes over time
- Based on Nmap scan parsed to Amap
- Security LiveCDs (Backtrack and nUbuntu)
- Output
 - CSV
 - TABS
 - CSV parsed to HTML
 - Email (whole output, just the latest changes or both)

PBNJ 1.0 - Limitations

- Not efficient
 - does not use modules
 - does not use Nmap's XML output
- Stores data in a CSV file
 - User looks directly at the CSV file

PBNJ 2.0 - Redesign Plan

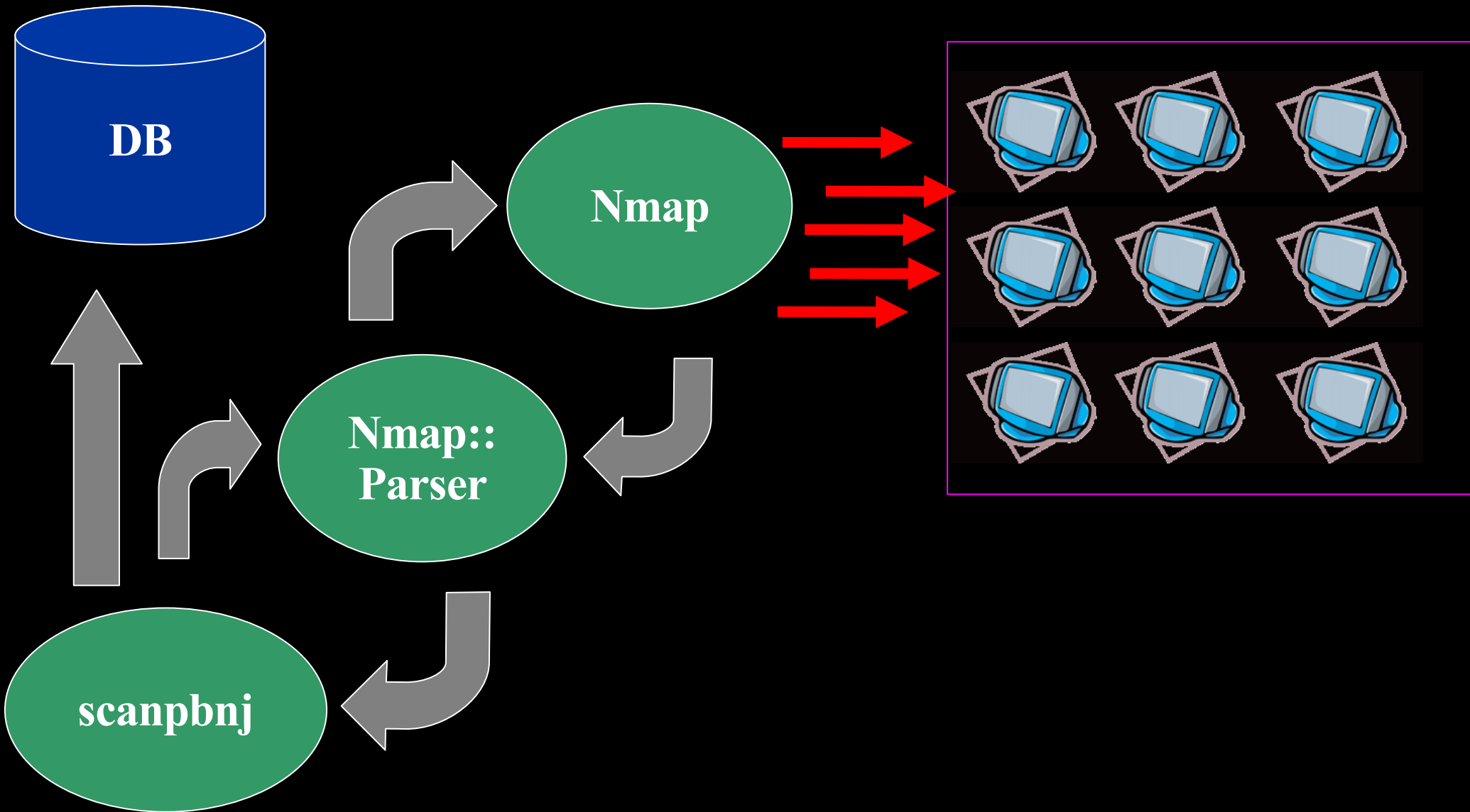


- Deprecate Amap (low number of dependencies)
- Store the information from the scans in database
- Flexible for user queries
- Parse XML rather than text

PBNJ 2.0 - Store Data in Database

- SQLite (File database)
 - doesn't require a real DB
 - won't have to worry about secure connection to DB
 - won't require a lot of effort to use
- User can dump the data elsewhere if needed
- Configurable for any DBI database

Scanner Functionality



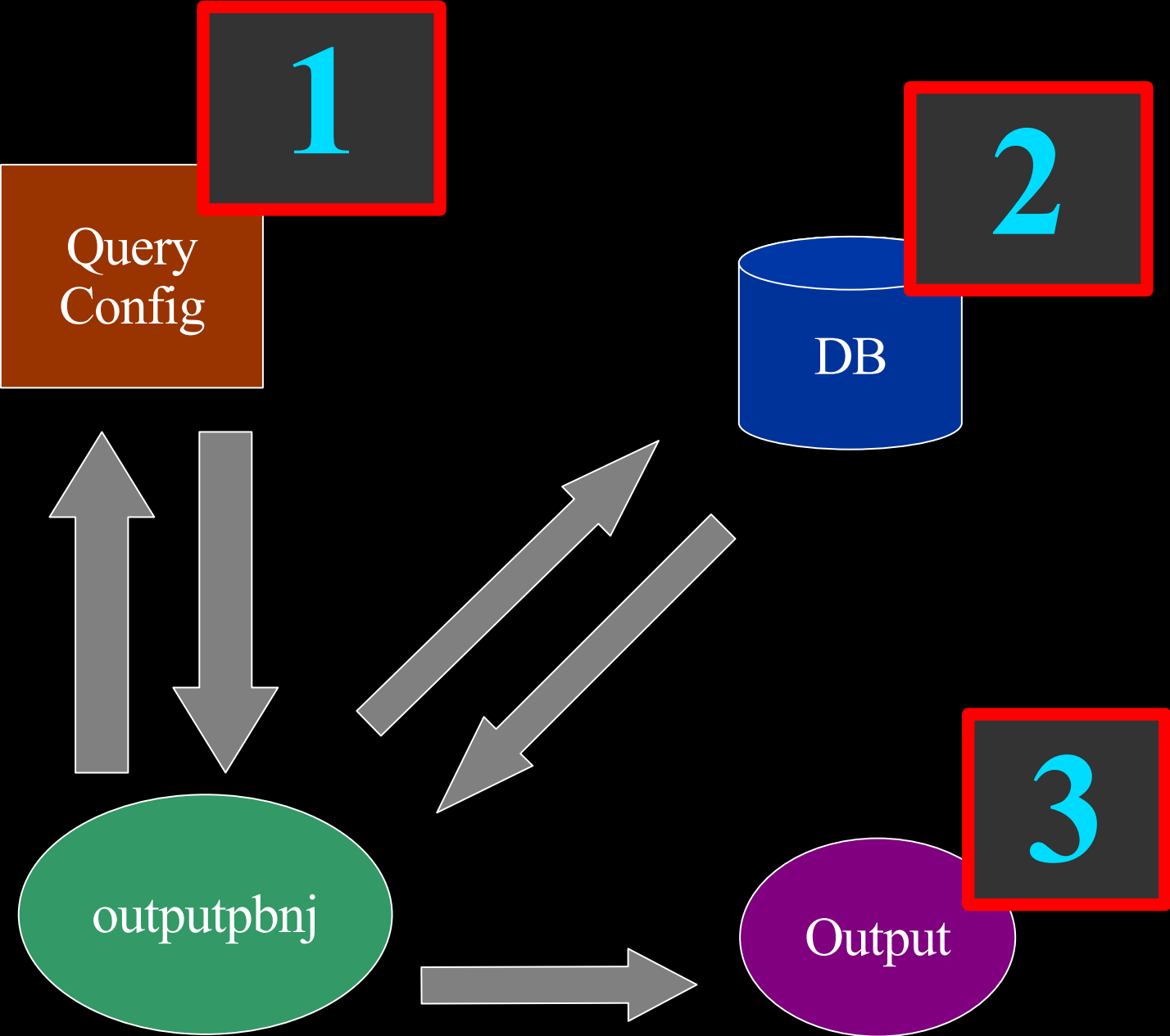
PBNJ 2.0 – Increased Flexibility

- User specified information
 - history of the scans
 - specific timespan
 - previous scan
- Used with other tools
 - develop other tools to process the data
 - develop other tools to parse the data

PBNJ 2.0 - Output the way you want it

- CSV
- TAB
- HTML
- Standard Output
- ... develop a module ...

Output Functionality



Query Configuration File

`- name: vulnssh`

Name of Query

Description

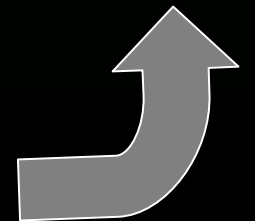
`desc: list all of the services that have old ssh running`

Query

`sql: |-`

```
select S.updated_on,M.ip,S.service,S.port,S.version from services as S,  
machines as M where service='ssh'and state='up' and version!='4.1p1'
```

Version to compare



PBNJ 2.0 – Easily extract the data you need

- User wants specific information
 - develop a SQL query
 - use popular SQL queries
- Have only the information you want in output
- Transfer data to real database(e.g. mysql)

Scan – Insert Host

```
$ sudo scanpbnj 127.0.0.1
```

Inserting Machine
Running
SSH and SMTP

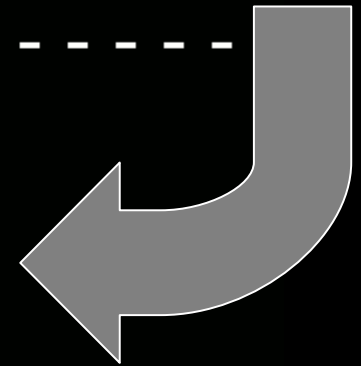
```
-----  
Starting Scan of 127.0.0.1
```

```
Inserting Machine
```

```
Inserting Service on 22:tcp ssh
```

```
Inserting Service on 25:tcp smtp
```

```
Scan Complete for 127.0.0.1  
-----
```

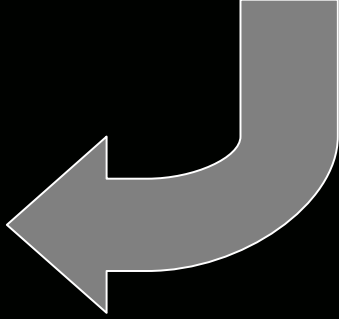


Scan – Without Change

**No State or
Version/Product
Changes**

```
$ sudo scanpbnj 127.0.0.1
```

```
-----  
Starting Scan of 127.0.0.1  
Machine is already in the database  
Checking Current Services  
    = ssh:22 is (4.2p1 Debian 7ubuntu3) OpenSSH  
    = smtp:25 is (unknown version) Postfix smtpd  
Scan Complete for 127.0.0.1  
-----
```



Output - Database Process Demo



Output of Query

```
$ ./outputpbnj --header -q vulnssh -t csv
updated_on,ip,service,port,version
Sun Jul 9 23:26:57 2006,127.0.0.1,ssh,22,4.2p1 Debian 7ubuntu3
```



Output to file

```
$
$
$
$ ./outputpbnj --header -q vulnssh -t csv --file out.pbnj

$ cat out.pbnj
updated_on,ip,service,port,version
Sun Jul 9 23:26:57 2006,127.0.0.1,ssh,22,4.2p1 Debian 7ubuntu3
```

Scan - Service Change

Stop Service

```
$ sudo /etc/init.d/ssh stop
* Stopping OpenBSD Secure Shell server... [ ok ]
$
$
$
$ sudo ./scanpbnj 127.0.0.1
```

Service State Down

```
-----
Starting Scan of 127.0.0.1
Machine is already in the database
Checking Current Services
    ! Service 22:tcp ssh is down
    = smtp:25 is (unknown version) Postfix smtpd
Scan Complete for 127.0.0.1
-----
```

Output - Latestinfo Query

```
$ ./outputpbj -q latestinfo
```

```
Wed Jul 12 22:24:05 2006      localhost      ssh      down      4.2p1 Debian 7ubuntu3      tcp
```

Hostname

Protocol

State

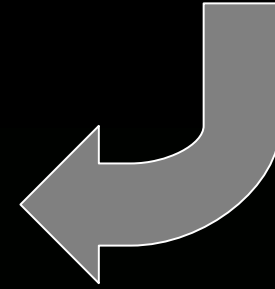
Service

Version

Date of Change

Scan – Nmap XML

Input
Nmap XML file



```
$ ./scanpbnj -x nmap.xml
```

```
-----  
Starting Scan of 127.0.0.1
```

```
Inserting Machine
```

```
Inserting Service on 22:tcp ssh
```

```
Inserting Service on 25:tcp smtp
```

```
Scan Complete for 127.0.0.1  
-----
```

Shell Script for Alerting

```
#!/bin/bash
# PBNJ 2.0 script to only send an email when a new change occurs
DIR=/root/data
CHANGE=change.out
TMP=tmp.out
USER=root
SUBJECT="[PBNJ] Latestinfo Alert `date`"
# sends the changes in email to the user
send_mail() {
    mv $TMP $CHANGE
    cat $CHANGE | mail -s "$SUBJECT" "$USER"
}
mkdir -p $DIR
cd $DIR
scanpbnj 192.168.10.0/24 > /dev/null 2> /dev/null
outputpbnj -q latestinfo -t csv > $TMP 2> /dev/null
if [ -e $CHANGE ];
then
    diff $CHANGE $TMP > /dev/null
    if [ $? -ne 0 ];
    then
        send_mail
    fi
else
    send_mail
fi
```

Set Proper Privs

- Make sure the file is executable:

```
$ sudo chmod +x /root/bin/alert_changes.sh
```

Add Entry to Crontab

- We then add the script to the Cron scheduler.

scan of the 10 network every 2 hours

#m	h	dom	mon	dow	user	command
----	---	-----	-----	-----	------	---------

16	*/2	*	*	*	root	/root/bin/alert_changes.sh
----	-----	---	---	---	------	----------------------------

Scenario – Discovery

- Scheduled Scans of a Range
- All machines running only SSH
- Rogue FTP Service
- Service or Host Discovery



Scenario – Monitor



- Scheduled Scans of Localhost
- Runs web server
- Notice Web server crashes
- Monitor Local or Remote Systems

Demo PBNJ 2.0

Available Today

- PBNJ – a suite of tools to monitor changes on a network over time
- <http://pbnj.sourceforge.net>
- Version 2.0 available today!
- Version 1.0 still available



- <http://www.samag.com/documents/s=10112/sam0702a/0702a.htm>
- http://pbnj.sf.net/scripts/alert_changes.sh

Install PBNJ with Package Management

- Debian (as root)
 `apt-get install pbnj`
- Gentoo (as root)
 `emerge pbnj`
- FreeBSD (as root)
 `cd /usr/ports/security/pbnj`
 `make install clean`

Q/A

References

- Fyodor, “Remote OS detection via TCP/IP Stack FingerPrinting”, June 2002
- Arkin Ofir, “ICMP Usage in Scanning” Version 3.0, June 2001
- Skoudis Ed, “Counter Hack”, Prentice Hall 2002
- Emailing a text-message to a phone
 - <http://www.livejournal.com/tools/textmessage.bml?mode=details>

PBNJ 2.0 - Schema

```
sqlite> .schema
CREATE TABLE machines (
    mid INTEGER PRIMARY KEY AUTOINCREMENT,
    ip TEXT,
    host TEXT,
    localh INTEGER,
    os TEXT,
    machine_created TEXT,
    created_on TEXT);
CREATE TABLE services (
    mid INTEGER,
    service TEXT,
    state TEXT,
    port INTEGER,
    protocol TEXT,
    version TEXT,
    banner TEXT,
    machine_updated TEXT,
    updated_on TEXT);
CREATE TABLE sqlite_sequence(name,seq);
```

Machine Table

- changes mean a new machine

Service Table

- state changes
- version changes
- banner changes

Thank You for Coming!



