Change Management:

**Dynamic Network Mapping**

LinuxWorld San Francisco  |  Security Track

Presented by  Joshua D. Abraham

August 16\textsuperscript{th} 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)
Fingerprinting

• 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

- 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

$ 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
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- Out of date instantly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)

<table>
<thead>
<tr>
<th></th>
<th>test1.out</th>
<th>test2.out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Starting Nmap 4.20 (<a href="http://insecure.org">http://insecure.org</a>) at 2007-01-16</td>
<td>Starting Nmap 4.20 (<a href="http://insecure.org">http://insecure.org</a>) at 2007-01-16</td>
</tr>
<tr>
<td>2</td>
<td>Interesting ports on localhost (127.0.0.1):</td>
<td>Interesting ports on localhost (127.0.0.1):</td>
</tr>
<tr>
<td>3</td>
<td>Not shown: 16 closed ports</td>
<td>Not shown: 16 closed ports</td>
</tr>
<tr>
<td>4</td>
<td>PORT STATE SERVICE VERSION</td>
<td>PORT STATE SERVICE VERSION</td>
</tr>
<tr>
<td>5</td>
<td>22/tcp open ssh OpenSSH 4.2p1 Debian 7ubuntu3.1 (pro</td>
<td>22/tcp open ssh OpenSSH 4.2p1 Debian 7ubuntu3.1 (pro</td>
</tr>
<tr>
<td>6</td>
<td>+ 25/tcp open smtp Postfix smtpd</td>
<td>25/tcp open smtp Postfix smtpd</td>
</tr>
<tr>
<td>7</td>
<td>631/tcp open ipp CUPS 1.2</td>
<td>631/tcp open ipp CUPS 1.2</td>
</tr>
<tr>
<td>8</td>
<td>Device type: general purpose</td>
<td>Device type: general purpose</td>
</tr>
<tr>
<td>9</td>
<td>Running: Linux 2.6.X</td>
<td>Running: Linux 2.6.X</td>
</tr>
<tr>
<td>10</td>
<td>OS details: Linux 2.6.14 - 2.6.16</td>
<td>OS details: Linux 2.6.14 - 2.6.16</td>
</tr>
<tr>
<td>11</td>
<td>Uptime: 0.985 days (since Mon Jan 15 18:49:22 2007)</td>
<td>Uptime: 0.985 days (since Mon Jan 15 18:49:22 2007)</td>
</tr>
<tr>
<td>12</td>
<td>Network Distance: 0 hops</td>
<td>Network Distance: 0 hops</td>
</tr>
<tr>
<td>13</td>
<td>Service Info: OS: Linux</td>
<td>Service Info: Host: oreo; OS: Linux</td>
</tr>
<tr>
<td>14</td>
<td>OS and Service detection performed. Please report any inc</td>
<td>OS and Service detection performed. Please report any inc</td>
</tr>
<tr>
<td>15</td>
<td>Nmap finished: 1 IP address (1 host up) scanned in 8.390</td>
<td>Nmap finished: 1 IP address (1 host up) scanned in 8.314</td>
</tr>
</tbody>
</table>
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?
"Not bad but I want to be notified via email"
$ diff -u test1.out test2.out | grep -Eo \n"((^\+[0-9].*)||(^-[0-9].*))" | \nmail "Changes `date`" root
• 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 Tunneled 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
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
- \ldots develop a module \ldots
Output Functionality

Query Config → 1 → DB → 2 → outputpbnj → 3 → Output
Name of Query

- name: vulnssh

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

© Joshua D. Abraham 2006. All Rights Reserved
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

```bash
$ sudo scanpbunj 127.0.0.1
```

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
$ 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

No State or Version/Product Changes
$ ./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

$ ./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

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

---------------------------------------------
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
---------------------------------------------
$ ./outputpbnj -q latestinfo

Wed Jul 12 22:24:05 2006  hostname  ssh  down  4.2p1 Debian 7ubuntu3  tcp
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
Available Today

- PBNJ – a suite of tools to monitor changes on a network over time
- Version 2.0 available today!
- Version 1.0 still available

- http://pbnj.sf.net/scripts/alert_changes.sh
Install PBNJ with Package Management

- Debian (as root)
  
  \texttt{apt-get install pbnj}

- Gentoo (as root)
  
  \texttt{emerge pbnj}

- FreeBSD (as root)
  
  \texttt{cd /usr/ports/security/pbnj}
  \texttt{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
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);
Thank You for Coming!